# Técnicas de Aprendizado de Máquina aplicados a Dicalcogenetos de Metais de Transição

**Guilherme dos Santos Marcon**

# Técnicas de Aprendizado de Máquina aplicados a Dicalcogenetos de Metais de Transição

**Guilherme dos Santos Marcon**

*Supervisor:* Juarez L. F. Da Silva

Monografia de conclusão de curso apresentada ao Instituto de Ciências Matemáticas e de Computação da Universidade de São Paulo para obtenção do título de Bacharel em Ciências de Computação.

**USP - São Carlos**
**Novembro de 2019**

# Dedicatória

Aos meus pais, Eliane e Nivaldo.

# Agradecimentos

Agradeço ao Professor Dr. Juarez L. F. da Silva pela orientação e oportunidade de realizar este projeto. Ao grupo QTNano pelo acolhimento e ensinamentos, em especial ao Naidel A. M. S. Caturello pela discussão sobre os dicalcogenetos, bem como o auxílio na escrita deste documento. Ao Professor Dr. Marcos G. Quiles pelas orientações relacionadas às técnicas de aprendizado de máquina.

Agradeço a Fundação de Apoio à Universidade de São Paulo (FUSP) pelo financiamento deste projeto.

Também gostaria de agradecer minha família e amigos pelo suporte.

# Resumo

Dicalcogenetos de Metais de Transição (DMTs) são compostos químicos definidos pela equação $MQ_2$, onde M é um metal de transição, como molibdênio (Mo) e tungstênio (W), e Q pode ser enxofre (S), telúrio (Te) ou selênio (Se). Eles possuem uma grande gama de morfologias, dimensionalidades e aplicações e, devido a essas propriedades, tem sido objeto de vários estudos desde a década de 1960, com o intuito de compreender melhor suas propriedades. No entanto, um empasse nesses estudos é o requisito computacional para se calcular essas propriedades, pode-se levar dias para que terminem. Neste quesito, o aprendizado de máquina é importante para a química pois, se for possível encontrar um erro baixo aceitável, os milissegundos de predição das ditas propriedades aceleraria em muito os estudos dessas estruturas. Portanto neste projeto, foi utilizado diferentes modelos de regressão e variações da Matriz de Coulomb para prever a energia total das moléculas de DMTs. Como resultado, a regressão Linear com os autovalores da Matriz de Coulomb ordenada obteve um erro médio de 3.31e-5%, equivalente à um erro normalizado de 6.3kcal/mol, um erro elevado impossível de substituir os cálculos atuais, que possuem um erro de 1kcal/mol. Concluindo que a Matriz de Coulomb não é o suficiente para gerar os erros baixos esperados, sendo necessário uma representação mais robusta.

**Palavras-chaves:** Matriz de Coulomb, Dicalcogenetos de Metais de Transição, Aprendizado de Máquina.

# Sumário

Capítulo

# 1

# Introdução

Machine learning (ML) é um campo da inteligência artificial que concerne o aprendizado progressivo de máquina através de algoritmos estatísticos [1]. O desenvolvimento da área de ML se deu para mitigar problemas que surgiam de processos de triagem onerosos [2], baseados em tentativa e erro [3]. O advento de técnicas de ML, então, foi integrado a várias áreas que demandam a projeção de resultados com a observância de dados prévios, como na engenharia de processos e controle, além de vários campos de novos materiais para otimização de estocagem e geração de energia [1]. Neste sentido, a aplicação de técnicas de ML na área da química vem ganhando crescente atenção na literatura [4], tendo sido aplicada com sucesso em estudos de reações químicas, notadamente no sentido de predição de performance de dada reação e quais os produtos das reações químicas estudadas. A integração de técnicas de ML com áreas da química foi determinante para o desenvolvimento de pacotes de ML como o *python materials genomics* (Pymatgen) [5], Factsage [6] e Aflow. Desta maneira, se desenvolveu o panorama de pacotes de alto rendimento *high-throughput* (HT) para o desenvolvimento de pesquisas que exijam menor tempo para o desenvolvimento de áreas que possam ser industrialmente úteis no desenvolvimento de novos materiais funcionais [7].

Dentre os materiais funcionais que têm ganho grande destaque na literatura estão materiais bidimensionais (2D), e, em particular, dicalcogenetos de metais de transição bidimensionais (DMTs 2D) [8]. Os DMTs 2D são definidos pela equação química $MQ_2$, na qual $M$ é o metal de transição, como Mo, e $Q$ = S, Se, Te. A variedade de composições químicas e os diferentes ambientes de coordenação no entorno dos átomos metálicos (politipo), como o octaédrico (1T), octaédrico distorcido (1T′) e trigonal prismática (2H), faz com que DMTs 2D possuam grande variedade de propriedades estruturais [9], eletrônicas [10] e energéticas [4] que os candidatam para uma gama sem precedentes de aplicações, que vão de catálise (como na reação de evolução

do hidrogênio e de redução do $CO_2$) até o uso como dispositivos fotovoltaicos [11] de espessura atômica, uma vez que as camadas de DMTs 2D de metais do grupo 6 são compostas de planos metálicos entremeados por planos de $Q$, sendo estas camadas interagentes através da ação de interações de van der Waals [7].

Em particular, DMTs baseados em molibdênio (Mo) possuem em sua carta de compostos o $MoS_2$, DMT 2D mais popularmente estudado na literatura [7], que tem encontrado validações teóricas e experimentais para os mais diversos usos de DMTs 2D. Por outro lado, o $MoTe_2$ tem sido o DMT 2D baseado em Mo com o menor número de estudos reportados na literatura [12], revelando, no entanto, propriedades distintas de suas contrapartes, servindo de base para o estudo de "novas físicas", como para semimetais de Weyl [13], isto é, materiais que possuem carreadores de carga que se comportam como partículas sem massa no ponto no qual as bandas de condução e valência se cruzam na forma de uma dispersão linear (nodo de Weyl). DMTs baseados em tungstênio (W), como o $WS_2$, são particularmente importantes devido ao maior raio atômico do W quando comparado com o Mo, o que permite que DMTs com W na composição sejam mais flexíveis em relação à modulação de propriedades físico-químicas através de dopagem [14], além de serem aplicáveis para materiais com aplicação em fotônica [15], fotocatálise [16] e dispositivos com alta pseudocapacitância [17].

Para que se possa compreender a físico-química básica de DMTs 2D é necessário que se obtenha conhecimento em nível atomístico sobre estes sistemas. Tal conhecimento detalhado é capaz de prover princípios de *design* [18] DMTs 2D energeticamente estáveis e que possam servir como materiais funcionais [7]. Para a avaliação de propriedades físico-químicas de DMTs 2D têm sido utilizada como ferramenta de cálculo o formalismo da teoria do funcional da densidade (DFT) [19], implementado na forma da resolução auto-consistente das equações de Kohn–Sham (KS) [20] em códigos computacionais que diferem na representação dos orbitais de KS. A resolução das equações de KS traz consigo uma escala de tempo que varia entre dias ou semanas dependendo do tamanho do sistema de DMT 2D de tamanho finito (nanofloco) investigado.

Ademais, as propriedades físico-químicas de nanoflocos de DMTs 2D dependem sensivelmente da geometria do nanofloco [12], o que faz com que a predição de propriedades físico-químicas destes sistemas precise envolver amostragens de configurações. Experimentalmente, o método de síntese mais utilizado para nanoflocos de DMTs 2D, *chemical vapor deposition*, CVD [21], permite a síntese de nanoflocos de distintas geometrias com politipos distintos, fato que integra predições teóricas com a possibilidade de síntese de compostos previamente investigados através da DFT. Tendo em vista o aumento da agilidade da pesquisa de nanoflocos de DMTs 2D, diminuindo custo e tempo computacional, isto é, diminuir a escala de tempo de dias para tempos irrisórios, na escala de minutos ou segundos, passa pela integração adequada de abordagens de ML aplicadas a estes sistemas [1].

Patra *et al.* [22] combinaram algoritmos genéticos (GAs) para a geração de estruturas com vacâncias de S de monocamadas de $MoS_2$ otimizadas com dinâmica molecular (DM) para compreender os mecanismos de extensão e coalescência de defeitos neste material em escalas de

tempo de minutos. Na investigação foi possível determinar a formação da fase 1T no entorno das vacâncias de enxofre, diferindo do politipo de monocamada estudada, 2H. Além do mais, os resultados obtidos através da combinação ML/DM foram validados através de microscopia eletrônica. Zhao *et al.* [1] aplicaram o procedimento de aplicação de métodos de amostragem e seleção de DMTs que encontrassem uso na detecção de $Hg^0$ baseados em ML, como implementado no Pymatgen, para selecionar previamente os materiais que seriam avaliados através de cálculos DFT. Portanto, os métodos de ML serviram como base de pré-processamento de seleção de materiais que poderiam encontrar as propriedades desejadas. Estabeleceu-se que o DMT com maior capacidade de detecção de $Hg^0$ foi o $NiS_2$.

Tawfik *et al.* [3] realizaram a predição de propriedades físico-químicas de bicamadas de DMTs 2D, encontrando a diminuição de cinco vezes no tempo de tratamento entre as técnicas aplicadas e os cálculos de DFT. Neste estudo, utilizou-se um conjunto de treinamento de 267 estruturas calculadas por DFT em um conjunto de testes de 1500 bicamadas, sendo o conjunto de testes provindo de Miro *et al.* [23]. Ademais, com os resultados obtidos, os autores ressaltam ser possível estudar 1,7 milhões de possibilidades de bicamadas de DMTs através das estruturas reportadas por Mounet *et al.* (dentro de c), que reporta 1800 blocos de construção de bicamadas para materiais 2D.

No entanto, não há na literatura nenhum estudo que combine técnicas de ML para o estudo de amostragens de nanoflocos de DMTs 2D. Para explorar estes aspectos, utilizamos conjuntos de DMTs no formato $(WQ_2)_n$, onde W = Mo ou W, Q = S, Se ou Te, $n = 1 - 16$ para os conjuntos com Mo e $n = 1 - 16$, 36, 66, 105 para os com W. Os modelos de ML para regressão, como Linear, Kernel Ridge e Redes Neurais Multi layer Perceptron [24], foram aplicados utilizando as representações obtidas através da Matriz de Coulomb [2, 25] para prever a energia total das moléculas.

O melhor modelo de regressão da energia total para estas moléculas com até 315 átomos foi o Linear, utilizando os autovalores da Matriz de Coulomb ordenada como representação. Foi obtido um erro médio absoluto de 3.31e-5%, que embora pareça baixo, ele se traduz para um erro médio normalizado por $n$ de 6.3kcal/mol. A DFT, para comparação, produz erros máximos de 1kcal/mol.

Concluiu-se que os autovalores não são o suficiente para gerar erros comparáveis à DFT. A regressão Linear é melhor baseada apenas no tamanho, gerando um erro médio normalizado de 5.5kcal/mol, também ainda insuficiente para reproduzir o comportamento de relaxação da DFT. É necessário uma representação mais robusta que os autovalores, por exemplo algo que contenha uma energia base para o tamanho da molécula além da estrutura.

Capítulo

# 2

# Metodologia

Para realizar o objetivo de aplicar modelos de aprendizado de máquinas nos DMTs e prever as suas propriedades de maneira mais rápida que a DFT, alguns passos são necessários: reunir, organizar e padronizar os dados existentes; representar as moléculas em um vetor de característica; selecionar os modelos para regressão, treiná-los e finalmente testá-los.

Para isso, será utilizada a linguagem Python [26], especificamente as bibliotecas: numpy, scikit-learn e keras. O Numpy [27] fornece diversas funções de manipulação de vetores, álgebra linear e outras operações matemáticas mais genéricas. O Scikit-Learn [28] é uma das bibliotecas mais utilizadas em Python para aprendizado de máquina, possuindo diversos métodos para mineração de dados e análise de dados. Por fim, o Keras [29] fornece uma interface de alto nível à biblioteca TensorFlow [30], que proveem rotinas de alto desempenho para treinamento de modelos de redes neurais.

## 2.1 Cronograma

Este trabalho teve início em Janeiro de 2019 como um projeto de Iniciação Científica. O cronograma então foi criado de maneira que o primeiro semestre fosse realizado atividades seguindo a mesma sequência de etapas aprendidas na disciplina SCC5871 de Aprendizado de Máquina. Veja a tabela 2.1 para mais detalhes.

**Tabela 2.1:** Atividades Planejadas

| Início | Fim | Descrição |
|---|---|---|
| 01/02/2019 | 15/02/2019 | Atividade 1: Familiarização com os dicalcogenetos |
| 15/02/2019 | 15/03/2019 | Atividade 2: Padronização dos Dados |
| 15/03/2019 | 15/04/2019 | Atividade 3: Estudo e uso da Matriz de Coulomb |
| 15/04/2019 | 01/10/2019 | Atividade 4: Estudo e uso dos algoritmos de Aprendizado de Máquina |
| 01/10/2019 | 01/11/2019 | Atividade 5: Análise dos resultados e escrita do TCC |
| 01/02/2019 | 30/12/2019 | Atividade 6: Participação nos workshops e seminários do grupo QTNano |

## 2.2 Dados

Os dados do projeto são formados por 6 conjuntos de treinamento, compostos por isômeros bidimensionais de tamanho finito (nanoflocos) e separados pela configuração molecular. Os conjuntos são formados por: (i) $(MoQ_2)_n$, $Q$ = S, Se, Te com $n$ = 1 – 16 e (ii) $(WQ_2)_n$, $Q$ = S, Se, Te com $n$ = 1 – 16, 36, 66, 105. Os conjuntos de (i) foram obtidos através do trabalho de Caturello *et al.* [12], enquanto que do trabalho de Da Silva *et al.* [18] foram obtidos os conjuntos utilizados em (ii). Para o conjunto de teste utilizou-se nanoflocos gerados através do método *drunk-walk* do próprio grupo QTNano.

Para cada uma dessas moléculas, existe um arquivo de saída do FHI-aims [31], o algoritmo utilizado para a calcular as energias totais dos isômeros, que utiliza o formalismo da teoria do funcional de densidade (DFT). O processo de avaliação de energia total é associado ao processo de relaxação estrutural, algoritmo que causa transformações na estrutura molecular em busca da mínima energia, como critério de parada, o processo é terminado quando a diferença de energia entre dois passos ou a força total atuando sobre os átomos do sistema é abaixo de um threshold. Uma vez que um mínimo de energia potencial é alcançado, a relaxação é finalizada. As propriedades físico-químicas dos sistemas otimizados em todos os passos de otimização dos cálculos são calculadas e estocadas nos arquivos de saída do programa.

A partir desse arquivo de saída, foi aplicado expressões regulares para extrair as informações, que são: a estrutura molecular e a energia total. A estrutura é salva em um arquivo no formato "M(n)Q(2*n)_numero-da-estrutura_id-da-iteração.xyz", contendo cada átomo da molécula seguido da sua posição no eixo X, Y e Z, e a energia (em elétrons-volt) de cada molécula é salva em uma tabela .csv.

O próximo passo é verificar a integridade desses dados, foi observado que existem átomos diferentes com a mesma energia, devido ao truncamento que o FHI-aims realiza na escrita do arquivo de saída, ou seja, as variações foram pequenas o suficiente para as energias truncadas serem iguais. Portanto, para que o conjunto se torne balanceado com energias únicas, foi retirado as duplicadas para moléculas de mesmo tamanho.

Em resumo, para cada conjunto, existe os arquivos .xyz contendo a estrutura de cada molécula dos conjuntos e uma tabela .csv com o nome da molécula e sua energia, onde todas as energias são únicas.

## 2.3 Representação molecular

Os algoritmos de aprendizado de máquina normalmente são compostos de dados padronizados para que todas as variáveis estejam na mesma ordem de grandeza. Para isso, é necessário então representar os dados de maneira relevante e de mesma dimensionalidade, chamado de vetor de característica. Neste projeto, esse vetor precisa representar tanto as moléculas de 3 átomos quanto as de 315.

Para isso, será utilizada a matriz de Coulomb [2] e suas variações: Ordenada e Autovalores [25]. A matriz de Coulomb gera uma representação matricial de uma molécula, levando em consideração os números atômicos e a posição de cada átomo. Ela é gerada a partir da equação:

$$\mathbf{M}_{\alpha\beta} = \begin{cases} 0,5Z_{\alpha}^{2,4} \text{ , se } \alpha = \beta \\ \frac{Z_{\alpha}Z_{\beta}}{|\mathbf{R}_{\alpha}-\mathbf{R}_{\beta}|} \text{ , se } \alpha \neq \beta \end{cases} , \tag{2.1}$$

na qual se utiliza as coordenadas cartesianas, $\{\mathbf{R}_{\alpha}\}$, enquanto que $Z_{\alpha}$ são as cargas nucleares dos $\alpha$-ésimos átomos componentes dos sistemas.

O tamanho da matriz não é invariável pelo da molécula, portanto ela é transformada em uma matriz MxM, onde M é o tamanho máximo das moléculas do conjunto, e os novos espaços são preenchidos com zero. Em sequência, é observado que a matriz é simétrica pela diagonal, portanto apenas a triangular superior é utilizada, achatando-a para finalmente considerar essa representação um vetor de característica.

As outras representações possuem passos extras: a matriz de Coulomb Ordenada é ordenada pela soma das linhas antes da separação da triangular superior; os Autovalores são literalmente os autovalores da matriz ordenada, gerando uma representação linear ao tamanho da molécula ao invés da quadrática das matrizes anteriores. Em resumo, os arquivos .xyz são transformados em vetores de características, existem aqueles quadráticos: a matriz original e a ordenada; e linear: os autovalores.

## 2.4 Modelos de Regressão

Possuindo os dados em vetores de características, é necessário escolher modelos de regressão para mapear a entrada para a saída desejada, isto é, conseguir a partir da estrutura molecular prever a energia total com rapidez. Como aprendizado de máquina não possui um modelo universal para se aplicar em todos os problemas, alguns foram escolhidos para serem utilizados

no projeto: Linear, Kernel Ridge, Rede Neural, K-Nearest Neighbors (KNN), Decision Tree e Random Forest [24, 32, 33].

### 2.4.1 Linear

Sendo uma das regressões mais simples, a Linear calcula um hiperplano que minimize a soma das distâncias entre a propriedade sendo prevista e o hiperplano. Se o problema for linearmente separável, esse modelo produz bons resultados. A regressão é dada pela fórmula:

$$Y = \beta_0 + \sum_{j=1}^{m} X_j \beta_j \tag{2.2}$$

no qual $Y$ é o valor da propriedade predita, $X$ é o vetor de característica, $m$ é o tamanho desse vetor e $\beta_j$ são os parâmetros ocultos da regressão.

### 2.4.2 Kernel Ridge Polinomial

Como uma escolha levemente mais complexa que o modelo anterior, o Kernel Ridge foi escolhido por conseguir resolver problemas não lineares. Utilizando-se do truque de kernel, que consiste em realizar transformações no vetor de característica (como aplicar uma exponencial), possibilita uma boa regressão para dados que não possuem comportamento linear.

### 2.4.3 Redes Neurais

Com uma maior complexidade, o principal modelo explorado foram as redes neurais, escolhidas pela variedade de customização. Veja a figura 2.1 para uma explicação.

O modelo Keras construído é composto de: até 2 camadas densas com ativações lineares, a quantidade de nós em cada uma varia de acordo com o vetor de característica, para a representação com matriz de Coulomb original e ordenada, que é quadrática, o modelo possui uma camada densa com 100 nós e outra com 1, já para os autovalores, os resultados fizeram o modelo evoluir até restar apenas uma camada densa com 1 nó, que faz o MLP se comportar de maneira parecida com uma regressão linear; o modelo então foi compilado utilizando loss=MAE, optimizer=default adam e metrics=MAPE.

### 2.4.4 Outros modelos

Para testes mais diversificados, foram escolhidos outros modelos para serem testados sem extensa exploração de parâmetros, que são: KNN, Decision Tree e Random Forest.

O KNN é um modelo que não necessita de treino em si, para cada dado novo, ele percorre todos os dados do conjunto de treino e encontra os K vizinhos mais próximos a partir de métricas como a distância euclideana, ele então retorna as médias das propriedades dos vizinhos como

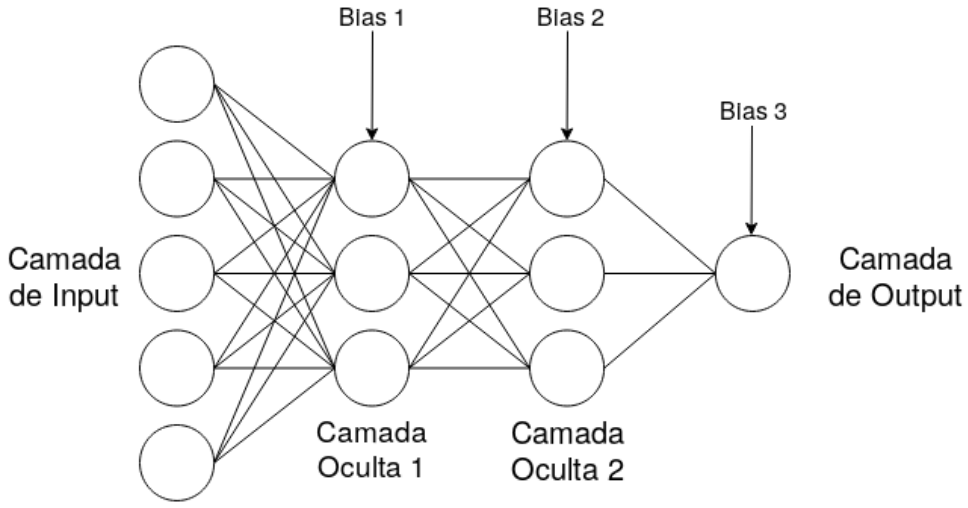


**Figura 2.1:** Exemplo de uma rede neural MLP, cada nó de camadas ocultas ou de saída são resultados de um valor bias com a soma dos valores da camada anterior multiplicados pelos respectivos pesos (marcados pelas linhas). O valor do nó então passa por uma função de ativação, como um efeito para simular a ativação de um neurônio. Então, comparando o resultado da camada de output com o valor dado de treino, os pesos e bias são atualizados de maneira a minimizar o erro.

predição. É um modelo que, dependendo da quantidade de dados para treino, as propriedades podem demorar para serem calculadas.

O Decision Tree e Random Forest são modelos baseados em árvore. No Decision Tree, a partir de um nó raiz, é calculado com os dados de treino algo que maximize o "ganho de informação". Basicamente, o nó divide em dois (ou mais) subconjuntos que possua uma diferença marcante. Cada divisão então se torna outro nó, no qual é calculado o ganho de informação, dividindo o subconjunto novamente, de maneira recursiva. A Random Forest nada mais é que várias Decision Trees, diminuindo a dependência em uma única árvore. Dependendo dos parâmetros utilizados, o modelo pode ser tão bem treinado para os dados de treino que gera um erro alto para os dados de teste (chamado de *overfitting*), o que não é algo desejado.

### 2.4.5 Validação e Métrica de Erro

Para o treino e validação dos métodos aplicados, utilizou-se a validação cruzada estratificada com 10 divisões, que foram estratificadas pelo tamanho das moléculas ($n$). Isso garante que cada divisão possui a mesma proporção de cada tamanho, diminuindo a variância do erro de validação.

Para a amostragem de erro, ao invés do Mean Absolute Error (MAE) utilizado na literatura, será utilizado o Mean Absolute Percentage Error (MAPE). Essa mudança se deve pelo tamanho máximo das moléculas, enquanto os conjuntos amplamente explorados na literatura são divisões do QM9 [34], que são moléculas orgânicas com até 9 átomos "pesados" (C, N, O, S), as moléculas deste projeto chegam a ter até 315 átomos. Quanto maior o tamanho de uma

molécula, maior sua energia total, portanto o MAE resultaria em um erro médio aparentemente inaceitável para moléculas com tamanho pequeno, mas utilizando a porcentagem é mais seguro afirmar a acurácia dos modelos.

Capítulo

# 3

# Desenvolvimento e Resultados

## 3.1 Dados

Após a extração dos dados únicos, os conjuntos de dados para treino são compostos dos seguintes valores:

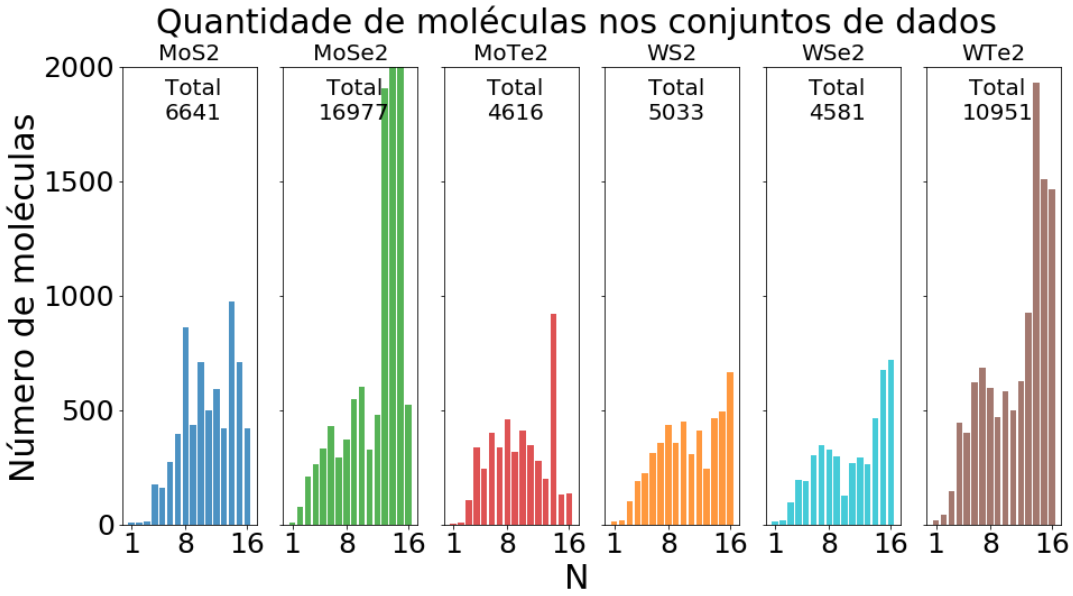


**Figura 3.1:** Configuração dos conjuntos $(MQ_2)_n$, com $n = 1 - 16$.

Adicionalmente, para que seja realizado testes com estruturas com $n \geq 16$, foram adicionadas novas moléculas para os conjuntos W$Q_2$, com $n = 36$, 66 e 105.

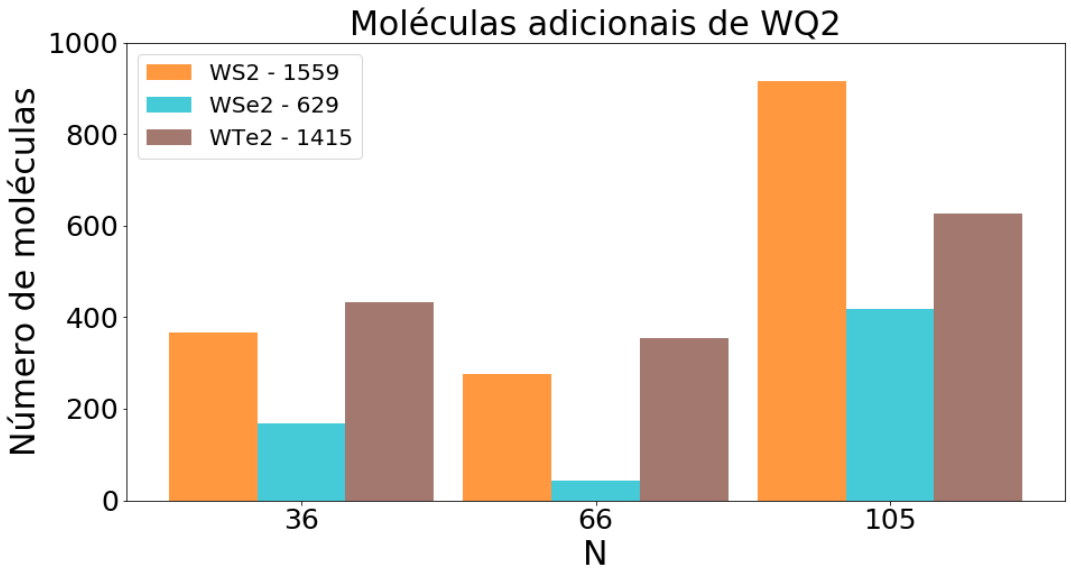


**Figura 3.2:** Quantidade de moléculas adicionais para os treinos e testes com $n \geq 16$. Na legenda, o número após o conjunto é o total de moléculas daquele grupo.

## 3.2 Representação via matriz de Coulomb

Utilizando o $MoTe_2$ como base, por ser o conjunto com menor quantidade de dados (espera-se que os outros conjuntos tenham uma melhor performance pelo maior número de moléculas), a escolha de qual representação da matriz de Coulomb a ser utilizada foi direta.

**Tabela 3.1:** MAPE das diferentes representações das moléculas do conjunto $MoTe_2$.

| Representação | Algoritmo de Regressão | MAPE |
|---|---|---|
| Original | Linear | 145.41 |
| Original | MLP | 1.86e-1 |
| Ordenada | Linear | 2.31e-4 |
| Ordenada | MLP | 5.08e-2 |
| Autovalores | Linear | 4.32e-5 |
| Autovalores | MLP | 2.24e-4 |

Observando os resultados da tabela 3.1 acima, pode-se verificar que, do ponto de vista do aprendizado de máquina, tanto para a regressão linear quanto para a MLP, os autovalores resultaram no menor erro.

## 3.3 Redes Neurais

Devido a customização extensa que as redes neurais possuem, vários testes foram realizados até o modelo final ser concretizado em um que utilize os melhores parâmetros encontrados.

### 3.3.1 Optimizers

A biblioteca Keras possuem uma série de "optimizers", heurísticas responsáveis por definir parâmetros importantes das redes neurais, como a taxa de aprendizado, momento e outros específicos de cada heurística.

**Tabela 3.2:** Exploração dos optimizers. Parâmetros especificados são aqueles que, no treino com menor MAPE, se diferencia do valor padrão.

| **Optimizer** | **MAPE** |
|---|---|
| Adadelta() | 0.0149 |
| SGD() | 0.0033 |
| Adamax(lr=0.01) | 0.0032 |
| Adagrad(lr=0.3) | 0.0026 |
| RMSprop() | 0.0021 |
| Nadam(schedule_decay=0.001) | 5.18e-4 |
| Adam() | 3.41e-4 |

A partir desses resultados, o optimizer Adam default foi utilizado para os diversos treinos.

### 3.3.2 Configuração das camadas

A princípio, a escolha da quantidade de camadas e de nós partiu do pressuposto: muitas camadas gera complicações para a regressão e que uma camada inicial com uma quantidade nós similar ao tamanho do vetor de característica é um bom ponto de partida.

Portanto, a partir dessas inferências, os primeiros modelos com a matriz de Coulomb original, cuja representação para moléculas com até 48 átomos gera um vetor de característica de 1176 atributos, foram criados com 2 camadas com ativações lineares: uma com 1000 nós e outra com 1 (a de saída), que para o conjunto $MoTe_2$ gerou um MAPE de 3.84e-1. Testes então foram realizadas até se chegar no modelo final: uma camada com 100 nós e outra com 1, gerando o MAPE de 1.86e-1. Para a matriz de Coulomb ordenada, esse modelo foi o mesmo, com um MAPE de 5.08e-2.

Para os autovalores da matriz de Coulomb, o modelo sofreu drásticas mudanças, como explicado na tabela 3.3, o modelo final, com apenas uma camada com ativação linear e 1 nó, é basicamente uma regressão linear.

**Tabela 3.3:** Evolução dos modelos MLP com a representação sendo os autovalores da Matriz de Coulomb. Os dados utilizados são os do conjunto $MoTe_2$. Nos Modelos, cada número significa a quantidade de nós em cada camada daquele modelo.

| Modelo | MAPE |
|---|---|
| 100, 1 | 3.73e-2 |
| 50, 1 | 1.84e-2 |
| 25, 1 | 1.62e-2 |
| 1, 1 | 7.05e-4 |
| 1 | 2.24e-4 |

## 3.4 Resultados dos diversos algoritmos de regressão

A partir então da representação pelos autovalores e utilizando o conjunto $MoTe_2$ como base, foram aplicados os demais algoritmos de regressão: Linear; Rede Neural MLP; Kernel Ridge Regression com kernel polinomial 2 (chamado de P2) e polinomial 3 (P3); K Nearest Neighbors considerando os 5 mais próximos (5NN) e outro com apenas o mais próximo (1NN); Decision Tree com diferentes critérios para qualidade da divisão, com MAE, mean squared error (MSE) e Friedman MSE (FMSE); Random Forest também com diferentes critérios de divisão, MAE e MSE.

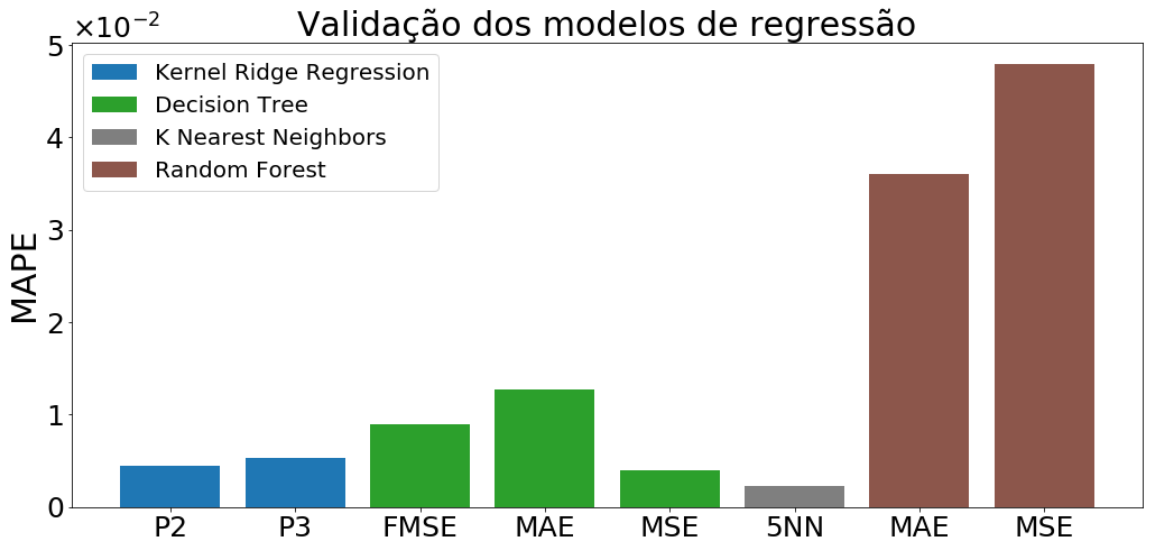


**Figura 3.3:** O MAPE dos diferentes algoritmos de regressão, utilizando o conjunto $(MoTe_2)_n$, $n = 1 - 16$.

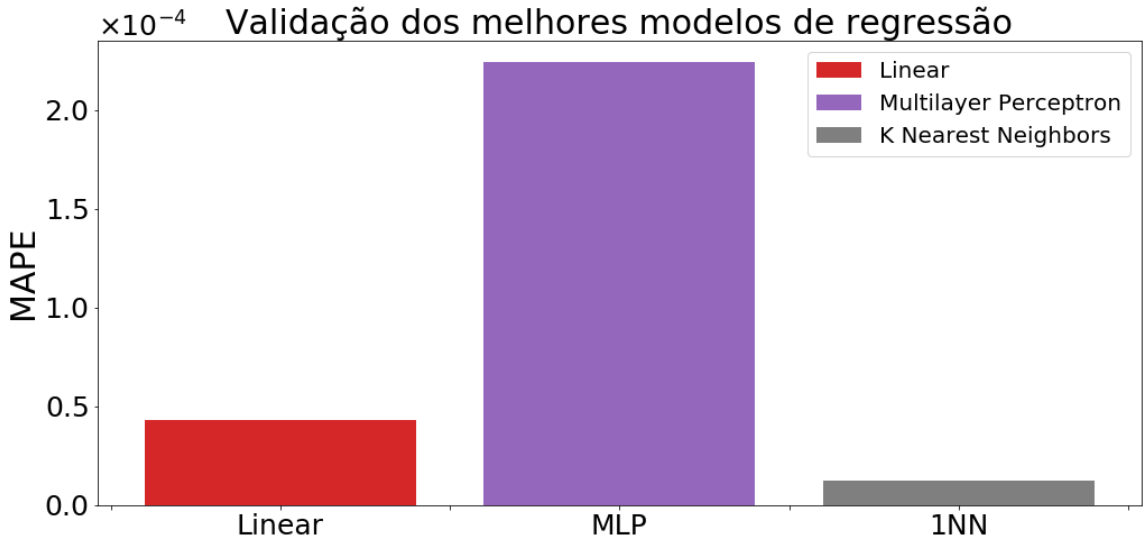


**Figura 3.4:** O MAPE dos 3 melhores algoritmos de regressão, utilizando o conjunto $(MoTe_2)_n$, $n = 1 - 16$.

## 3.5 Regressão da energia total

Aplicando os 3 melhores algoritmos (Linear, MLP e 1NN) para os demais conjuntos geram o seguinte resultado:

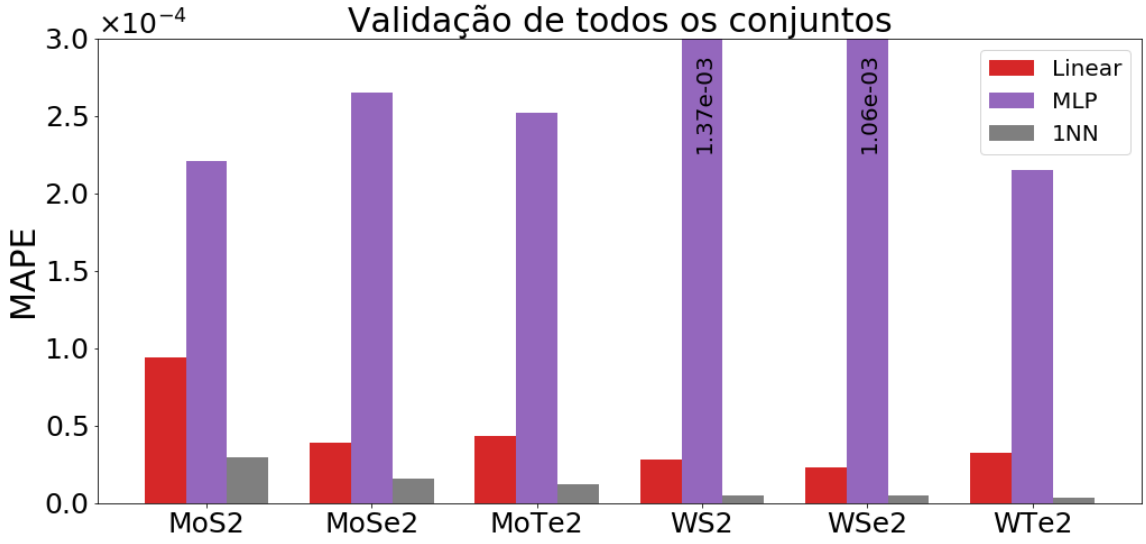


**Figura 3.5:** MAPE da energia total de todos os conjuntos $(MQ_2)_n$, $n = 1 - 16$.

O modelo com menor erro se tornou o 1NN, o vizinho mais próximo, apesar do erro ser menor que a metade do Linear, o 1NN não é necessariamente o melhor regressor, ele possui alguns problemas que será evidenciado no próximo passo: se houver intervalos de tamanhos que não

existem no conjunto de treino, o 1NN gera erros muito piores que os outros, já que ele atribui uma energia de uma molécula de tamanho diferente.

A rede neural MLP neste caso é uma regressão Linear com passos extras desnecessários, gerando um resultado que, com essa configuração, nunca vai ser melhor que a própria Linear.

A Linear em si é a melhor regressão, isso pode se dar pelo fato que todas as informações em torno das moléculas possuem comportamento linear: o formato das moléculas $(MQ_2)_n$ é linear pelo N, a representação com os autovalores, as energias das moléculas de treino (veja a figura 3.6). Todas essas características fazem a regressão Linear ser surpreendentemente o melhor modelo de regressão para esses tipos de DMTs.

## 3.6 Resultados com intervalos entre N

O último passo consiste em expandir o tamanho das moléculas, adicionando no treino as moléculas adicionais dos conjuntos $(WQ_2)_n$, $n$ = 36, 66, 105. As moléculas de N = 8 são também retiradas para utilizá-las apenas para teste, para que seja possível verificar o comportamento entre os modelos deste projeto e os cálculos da DFT, o modelo de relaxação atual que demora dias para ser computado.

Pela figura 3.6 e como evidenciado anteriormente, o modelo do vizinho mais próximo (1NN) gera valores para N do tamanho existente no conjunto, quanto existem tamanhos em falta, ele atribui energias de tamanhos diferentes, o que não é interessante para simular a relaxação da DFT.

A regressão linear e rede neural MLP conseguem realizar regressões relativamente boas para os intervalos, mas observando a figura 3.7, o erro ainda é muito grande comparado com a DFT, impossibilitando a simulação dos cálculos da DFT. Enquanto a linear possui um MAE normalizado por N de 6kcal/mol, a DFT consegue erros máximos de 1kcal/mol para qualquer valor de N.

## 3.7 Discussão dos Resultados

Observando os resultados, principalmente a figura 3.7, fica claro que certos modelos de regressão e representações moleculares não são o suficiente para substituir os cálculos de relaxação da DFT.

O modelo KNN não deve ser utilizado com esse objetivo, como o intuito é calcular as propriedades de moléculas que não possuem o tamanho no conjunto de treino, o resultado nunca seria satisfatório. E como o processo de relaxação em si causa pequenas mudanças nas moléculas, usar o KNN para substituir um método com essas características, além de poder resultar em energias de tamanhos diferentes, pode resultar em várias moléculas com leves variações terem a mesma energia prevista.

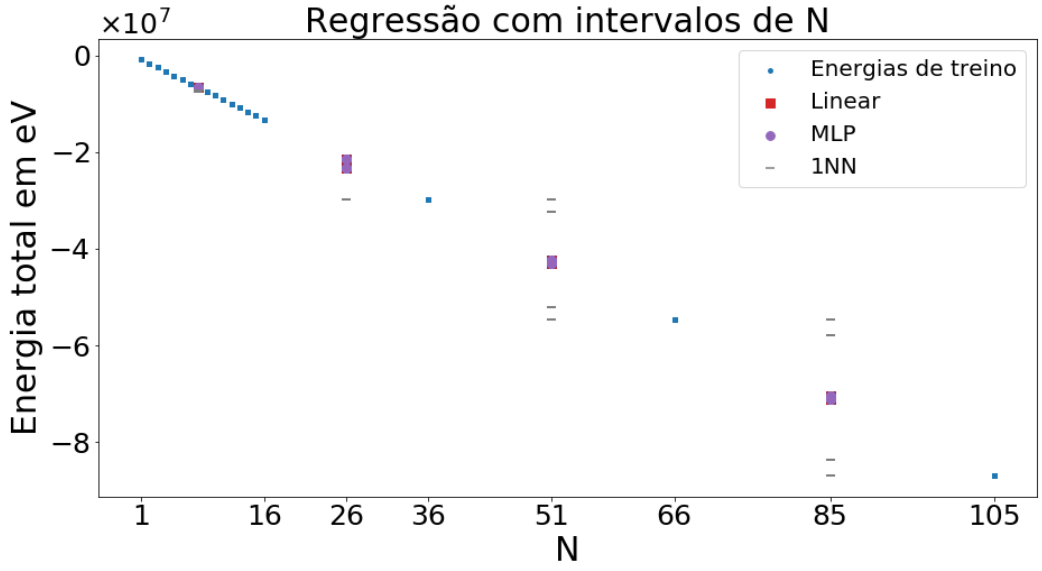


**Figura 3.6:** Resultado da regressão com o conjunto $(WTe_2)_n$, $n = 1 - 7, 9 - 16, 36, 66, 105$. Testada com moléculas de N = 8, 26, 51, 85.

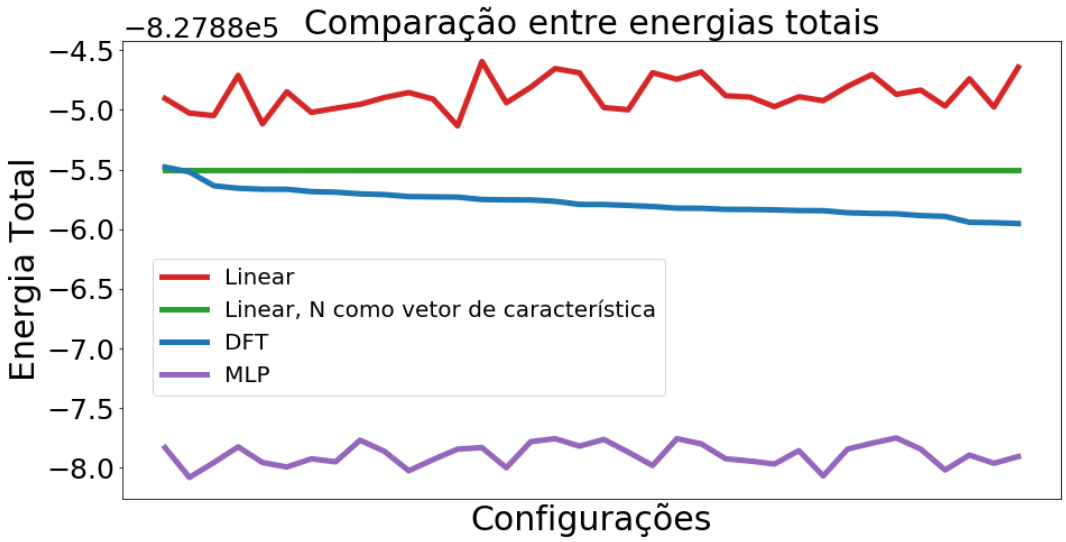


**Figura 3.7:** Comparação entre as energias totais resultantes dos cálculos de DFT, do modelo Linear e do MLP para $WTe_{2n}$, $n = 8$. As configurações são compostas apenas das moléculas pré-expansão dos dados, que são as moléculas finais da relaxação da DFT, ordenadas pela energia total da DFT.

Na questão do modelo linear, a regressão realizada está mais dependente do tamanho da molécula que a estrutura dela em si. Observando o comportamento das energias (figura 3.6), é possível observar claramente a relação entre o tamanho da molécula e a energia, algo que é óbvio do ponto de vista químico. Mas para o aprendizado de máquina, o modelo de regressão não "sabe" que deve utilizar a estrutura da molécula, é algo que o vetor de característica deve deixar explícito,

e apesar de os autovalores da matriz de Coulomb serem originados da estrutura molecular, o aspecto mais marcante que se pode obter deles é o tamanho da molécula.

A rede neural possui o mesmo problema do modelo anterior, os autovalores dão uma representação que o melhor modelo para fazer sua regressão é o linear, portanto faz sentido a rede neural tentar imitá-lo.

Analisando a figura 3.7, usar a regressão linear com apenas o tamanho da molécula como vetor de característica gera um erro menor que com os autovalores, mas ainda insuficiente para substituir a relaxação da DFT. Portanto para se obter um modelo capaz de substituir esses cálculos, é necessário uma representação molecular mais robusta para utilizar no aprendizado de máquina, algo que contenha informações explícitas sobre o tamanho, átomos, estrutura, talvez até usar como base a energia calculada pela regressão linear usando apenas o tamanho da molécula como vetor de característica, de modo que os outros atributos insiram as pequenas variações da relaxação.

Capítulo

# 4

# Conclusão

## 4.1 Discussões finais

Algoritmos de computação com mesma finalidade se diferenciam pelas permutações entre tempo de execução, uso de memória e acurácia. Os cálculos de DFT são extremamente precisos, com erro de no máximo 1kcal/mol, e mesmo com supercomputadores, o tempo de execução é enorme. O aprendizado de máquina, em contrapartida, demora milissegundos para predizer neste caso a energia, mas resulta em um maior erro, com uma média de 60kcal/mol para moléculas com até 315 átomos. O estudo para moléculas deste tamanho não são muito populares, portanto para melhor comparação, esse erro normalizado por N é em torno de 6.3kcal/mol.

Sobre a representação molecular, ela deve ser incrementada, um dos problemas é a falta de informação suficiente para diferenciar os isômeros dos DMTs. Visto que o menor erro foi utilizando apenas o tamanho da molécula como vetor de característica (figura 3.7), é necessário uma representação mais robusta que os autovalores da Matriz de Coulomb.

Sobre os modelos de regressão, o KNN se mostrou impróprio como um candidato para substituir a DFT e nenhum outro obteve um erro suficientemente baixo para simular a relaxação. Caso a representação molecular seja incrementada, seria necessário uma re-exploração dos modelos, principalmente das redes neurais.

Portanto, de forma geral, utilizar a Matriz de Coulomb original, ordenada ou seus autovalores como vetor de característica para os DMTs 2D não é o suficiente para gerar um erro baixo o suficiente para substituir o processo de relaxação da DFT.

## 4.2 Considerações sobre o curso de graduação

De modo geral, o curso de graduação me ensinou a ter independência, a pesquisar e aprender de maneira própria. A base ensinada pelas disciplinas iniciais, como Introdução a Ciência de Computação (ICC), foram essenciais para meu interesse nesse universo de programação, já que entrei na faculdade sem saber nada sobre.

Em específico para o projeto, as matérias de Aprendizado de Máquina e Redes Neurais foram fundamentais para o ponto de partida destas ideias, os trabalhos dessas disciplinas ensinaram a como organizar, tratar os dados e como começar a trabalhar com as redes neurais.

## 4.3 Sugestões para o curso de graduação

É impossível realçar o quão importante foi a disciplina de ICC, a extensa lista de exercício se transformou em um gosto por descobrir coisas novas e resolver problemas de programação. Portanto espero que o cuidado em como essas disciplinas iniciais são ministradas seja mantido.

## 4.4 Planos para o futuro

Ainda possuo outro semestre de estágio ou projeto de graduação para fazer, pretendo fazer um estágio e avaliar qual dos dois ambientes de trabalho prefiro.

# Referências


[1] Haitao Zhao, Collins I. Ezeh, Weijia Ren, Wentao Li, Cheng Heng Pang, Chenghang Zheng, Xiang Gao, and Tao Wu. Integration of machine learning approaches for accelerated discovery of transition-metal dichalcogenides as hg0 sensing materials. *Applied Energy*, 254:113651, 2019.

[2] Matthias Rupp, Alexandre Tkatchenko, Klaus-Robert Müller, and O. Anatole von Lilienfeld. Fast and accurate modeling of molecular atomization energies with machine learning. *Phys. Rev. Lett.*, 108(5):058301, Jan. 2012.

[3] Sherif Abdulkader Tawfik, Olexandr Isayev, Catherine Stampfl, Joe Shapter, David A. Winkler, and Michael J. Ford. Efficient prediction of structural and electronic properties of hybrid 2d materials using complementary dft and machine learning approaches. *Advanced Theory and Simulations*, 2(1):1800128, 2019.

[4] Manish Chhowalla, Hyeon Suk Shin, Goki Eda, Lain-Jong Li, Kian Ping Loh, and Hua Zhang. The Chemistry of Two-Dimensional Layered Transition Metal Dichalcogenide Nanosheets. *Nat. Chem.*, 5(4):263–275, apr 2013.

[5] Shyue Ping Ong, William Davidson Richards, Anubhav Jain, Geoffroy Hautier, Michael Kocher, Shreyas Cholia, Dan Gunter, Vincent L. Chevrier, Kristin A. Persson, and Gerbrand Ceder. Python materials genomics (pymatgen): A robust, open-source python library for materials analysis. *Computational Materials Science*, 68:314 – 319, 2013.

[6] P. Chartrand S. A. Decterov G. Eriksson A.E. Gheribi K. Hack I. H. Jung Y. B. Kang J. Melançon A. D. Pelton S. Petersen C. Robelin. J. Sangster C. W. Bale, E. Bélisle and M-A. Van Ende. Factsage thermochemical software and databases, 2010-2016. *Calphad*, 54:35 – 53, 2016.

[7] Sajedeh Manzeli, Dmitry Ovchinnikov, Diego Pasquier, Oleg V. Yazyev, and Andras Kis. 2D Transition Metal Dichalcogenides. *Nat. Rev. Mater.*, 2(8):17033–17047, jun 2017.

[8] Ping Cui, Jin-Ho Choi, Wei Chen, Jiang Zeng, Chih-Kang Shih, Zhenyu Li, and Zhenyu Zhang. Contrasting Structural Reconstructions, Electronic Properties, and Magnetic Orderings along Different Edges of Zigzag Transition Metal Dichalcogenide Nanoribbons. *Nano Lett.*, 17(2):1097–1101, jan 2017.

[9] Eric M. Vogel and Joshua A. Robinson. Two-Dimensional Layered Transition-Metal Dichalcogenides for Versatile Properties and Applications. *MRS Bull.*, 40(07):558–563, jul 2015.

[10] Kin Fai Mak and Jie Shan. Photonics and Optoelectronics of 2D Semiconductor Transition Metal Dichalcogenides. *Nat. Photon.*, 10(4):216–226, apr 2016.

[11] Deep Jariwala, Tobin J. Marks, and Mark C. Hersam. Mixed-dimensional van der waals heterostructures. *Nat. Mater.*, 16(2):170–181, aug 2016.

[12] Naidel A. M. S. Caturello, Rafael Besse, Augusto C. H. Da Silva, Diego Guedes-Sobrinho, Matheus P. Lima, and Juarez L. F. Da Silva. Ab Initio Investigation of Atomistic Insights into the Nanoflake Formation of Transition-Metal Dichalcogenides: The Examples of $MoS_2$, $MoSe_2$, and $MoTe_2$. *J. Phys. Chem. C*, 122(47):27059–27069, nov 2018.

[13] Youngdong Yoo, Zachary P. DeGregorio, Yang Su, Steven J. Koester, and James E. Johns. In-plane 2H-1T′ $MoTe_2$ homojunctions synthesized by flux-controlled phase engineering. *Adv. Mater.*, 29(16):1605461, feb 2017.

[14] Zhengyang Cai, Bilu Liu, Xiaolong Zou, and Hui-Ming Cheng. Chemical vapor deposition growth and applications of two-dimensional materials and their heterostructures. *Chem. Rev.*, 118(13):6091–6133, jan 2018.

[15] Sungwook Hong, Aravind Krishnamoorthy, Pankaj Rajak, Subodh Tiwari, Masaaki Misawa, Fuyuki Shimojo, Rajiv K. Kalia, Aiichiro Nakano, and Priya Vashishta. Computational synthesis of $MoS_2$ layers by reactive molecular dynamics simulations: Initial sulfidation of $MoO_3$ surfaces. *Nano Lett.*, 17(8):4866–4872, jul 2017.

[16] Yanfeng Chen, Jinyang Xi, Dumitru O. Dumcenco, Zheng Liu, Kazu Suenaga, Dong Wang, Zhigang Shuai, Ying-Sheng Huang, and Liming Xie. Tunable band gap photoluminescence from atomically thin transition-metal dichalcogenide alloys. *ACS Nano*, 7(5):4610–4616, apr 2013.

[17] Ali Eftekhari. Low Voltage Anode Materials for Lithium-Ion Batteries. *Energy Storage Materials*, 7:157–180, apr 2017.

[18] Augusto C. H. Da Silva, Naidel A. M. S. Caturello, Rafael Besse, Matheus P. Lima, and Juarez L. F. Da Silva. Edge, size, and shape effects on $WS_2$, $WSe_2$, and $WTe_2$ nanoflake stability: Design principles from an ab initio investigation. *Phys. Chem. Chem. Phys.*, 21(41):23076–23084, 2019.

[19] P. Hohenberg and W. Kohn. Inhomogeneous electron gas. *Phys. Rev.*, 136(3B):B864–B871, Nov. 1964.

[20] W. Kohn and L. J. Sham. Self-consistent equations including exchange and correlation effects. *Phys. Rev.*, 140(4A):A1133–A1138, Nov. 1965.

[21] Jeffrey D. Cain, Fengyuan Shi, Jinsong Wu, and Vinayak P. Dravid. Growth Mechanism of Transition Metal Dichalcogenide Monolayers: The Role of Self-Seeding Fullerene Nuclei. *ACS Nano*, 10(5):5440–5445, may 2016.

[22] Tarak K. Patra, Fu Zhang, Daniel S. Schulman, Henry Chan, Mathew J. Cherukara, Mauricio Terrones, Saptarshi Das, Badri Narayanan, and Subramanian K. R. S. Sankaranarayanan. Defect dynamics in 2-d mos2 probed by using machine learning, atomistic simulations, and high-resolution microscopy. *ACS Nano*, 12(8):8006–8016, 2018. PMID: 30074765.

[23] Pere Miró, Martha Audiffred, and Thomas Heine. An atlas of two-dimensional materials. *Chem. Soc. Rev.*, 43:6537–6554, 2014.

[24] Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *Elements of Statistical Learning*. Springer Series in Statistics, 2 edition, 2016.

[25] Grégoire Montavon, Katja Hansen, Siamac Fazli, Matthias Rupp, Franziska Biegler, Andreas Ziehe, Alexandre Tkatchenko, Anatole V. Lilienfeld, and Klaus-Robert Müller. Learning invariant representations of molecules for atomization energy prediction. In F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 25*, pages 440–448. Curran Associates, Inc., 2012.

[26] G. van Rossum. Python tutorial. Technical Report CS-R9526, Centrum voor Wiskunde en Informatica (CWI), Amsterdam, May 1995.

[27] Travis E Oliphant. *A guide to NumPy*, volume 1. Trelgol Publishing USA, 2006.

[28] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[29] François Chollet et al. Keras. `https://keras.io`, 2015.

[30] Martin Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, Manjunath Kudlur, Josh Levenberg, Rajat Monga, Sherry Moore, Derek G. Murray, Benoit Steiner, Paul Tucker, Vijay Vasudevan, Pete Warden, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. Tensorflow: A system for large-scale machine learning. In *12th USENIX Symposium on Operating Systems Design and Implementation (OSDI 16)*, pages 265–283, 2016.

[31] Volker Blum, Ralf Gehrke, Felix Hanke, Paula Havu, Ville Havu, Xinguo Ren, Karsten Reuter, and Matthias Scheffler. Ab Initio Molecular Simulations With Numeric Atom-Centered Orbitals. *Comp. Phys. Comm.*, 180(11):2175–2196, nov 2009.

[32] Christopher M. Bishop. *Pattern Recognition and Machine Learning*. Springer Series in Statistics, 2011.

[33] Simon Haykin. *Neural Networks and Learning Machines*. Pearson Prentice Hall, 3 edition, 2008.

[34] Zhenqin Wu, Bharath Ramsundar, Evan N. Feinberg, Joseph Gomes, Caleb Geniesse, Aneesh S. Pappu, Karl Leswing, and Vijay Pande. Moleculenet: A benchmark for molecular machine learning, 2017.