University of Amsterdam

Masters Thesis

Deep learning in quantum chemistry: Fock matrix level predictions for DFT based methods

Author:
Toby van Gastelen
Supervisor:
dr. Saad Yalouz

Examiner:
prof. dr. Lucas Visscher
Second assessor:
prof. dr. ir. Alfons Hoekstra

A thesis submitted in partial fulfilment of the requirements for the degree of Master of Science in Computational Science

in the

Computational Science Lab
Informatics Institute

August 2020

Declaration of Authorship

I, Toby van Gastelen, declare that this thesis, entitled 'Deep learning in quantum chemistry: Fock matrix level predictions for DFT based methods' and the work presented in it are my own. I confirm that:

- This work was done wholly or mainly while in candidature for a research degree at the University of Amsterdam.
- Where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated.
- Where I have consulted the published work of others, this is always clearly attributed.
- Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work.
- I have acknowledged all main sources of help.
- Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself.

Signed:

Date: 17 August 2020

UNIVERSITY OF AMSTERDAM

Abstract

Faculty of Science
Informatics Institute

Master of Science in Computational Science

Deep learning in quantum chemistry: Fock matrix level predictions for DFT based methods

by Toby van Gastelen

Machine learning is becoming increasingly applied to the field of quantum chemistry. Most of the existing approaches focus on the prediction of a subset of molecular properties from molecular geometries or results from less accurate methods. The application of such models can help to massively reduce the time required to determine molecular properties of interest, *e.g.* for drug-design or materials science, which typically involves going through large data bases of molecular structures. However, recently an alternative approach has come forward. Instead of predicting these properties, one can directly predict the converged Fock matrix encoding the electron-density from which these properties are derived in Kohn-Sham density functional theory (DFT). Based on this, we present the idea of using machine learning to obtain converged DFT Fock matrices by supplying an initial guess constructed from a superposition of free-atom densities. To explore the possibilities of such an approach, we first consider a single water molecule. We show that a single neural network is well capable of reproducing the ground state energy for the different conformations of water through reproducing the respective Fock matrices. After this we shift to density functional based tight binding (DFTB), and show that neural networks are capable of bridging the gap between first order DFTB1 and second order self-consistent charge (SCC)-DFTB by correcting the net Mulliken charges on each atom. For this we employ a data set containing $\sim$ 7000 small organic molecules and use rotationally invariant symmetry functions to describe the atomic environments. We find that this method generalizes well to molecules not contained in the training data. With this knowledge, we return to DFT with the aim of generalizing the Fock matrix predictions to small organic molecules. For this we construct a model containing fourteen separate neural networks and train them on an enhanced version of the same data set. We find that, although resulting Fock matrices are not accurate enough to serve as a final result, they can be used to speed up the self-consistent field procedure.

Acknowledgements

During this internship I have been fortunate enough to have received the help of many wonderful people, both inside and outside the university, some of whom I would like to address here personally. First of all, I would like to acknowledge the entire theoretical chemistry department of the Vrije Universiteit Amsterdam for the warm reception and the continuous support throughout this project. In particular I would like to thank:
Lucas Visscher for giving me the opportunity to do my master's project in his research group and for the numerous suggestions and ideas he provided to aid in our research and the writing of this thesis. Alfons Hoekstra for taking the time to take on the role of second assessor. Augusto Gerolin for helping me and my supervisor Saad get familiar with optimal transport theory, which unfortunately has yet to find its use for this type of machine learning applications. Thomas Soini and Robert Rüger for the assistance with regards to the DFTB part of this thesis. Furthermore, I am grateful to Emiel Koridon, Annina Lieberherr, Jakob Günther, Federico Mellini, and Bas van de Beek for the time spent and answering some of my "intelligent" questions. Lastly, I would like to express my gratitude towards my supervisor Saad Yalouz. In the first place, for the continuous support on both an academic and personal level, and for always taking the time to help me with any problems that came up along the way. Secondly, for proofreading my thesis numerous times and providing constructive feedback. Lastly, I am thankful for the perspective he has given me on science in general.

With regards to people outside of the university I would like to thank:
Ruben Konijn for taking the time out of his busy schedule to listen to my machine learning problems and provide some useful suggestions. I am especially grateful to my parents for providing me with the opportunity and encouragement to complete my studies. Finally, I would like to express my appreciation towards my girlfriend Sherylene Veira for supporting me during this process and helping me in any way she could.

Contents

List of Figures

List of Tables

Abbreviations

1RDM one-body reduced density matrix

BO Born-Oppenheimer

C.I. confidence interval

CC coupled-cluster

CCS coupled-cluster singles

CCSD coupled-cluster singles-doubles

CI configuration-interaction

CIS configuration-interaction singles

CISD configuration-interaction singles-doubles

DFT density functional theory

DFTB density functional based tight binding

GTO Gaussian-type orbital

HF Hartree-Fock

MAE mean absolute error

MO molecular orbital

MP Møller–Plesset perturbation theory

MSE mean squared error

NN neural network

PES potential energy surface

ReLU rectified linear unit

SCC self-consistent charge

SCF self-consistent field

SD Slater determinant

STO Slater-type orbital

Chapter 1

Introduction

Chemistry is the field that studies the properties of matter and their interactions. This knowledge is used to create new materials and molecules used in medicine, solar-cells, packaging *etc*. Considering that the reactions involved in the creation of these substances happens on such small time scales and the involved particles are incomprehensibly small, it is typically infeasible to study these interactions in detail in an experimental setting. Consequently, a different approach is required to study these molecular structures and their building-blocks. For this one has to resort to the the laws of quantum mechanics that govern the behaviour of the electrons and atomic nuclei that make up a molecule. Applying these laws to chemical systems is what concerns the field of quantum chemistry, which typically involves the use of computer simulations to study chemical processes. Quantum chemistry techniques are usually employed to predict properties such as the stability of different conformations of a molecule, the rate at which chemical reactions occur, and how different molecules interact. Obtaining these features requires quantifying the behaviour of the electrons which give rise to the electronic structure. However, solving this electronic structure problem requires a lot of computing power. This mostly comes from the complex electron-electron interactions which become increasingly more costly to deal with when the size of the system increases. It is therefore that many approximations are introduced to study larger molecular systems [4].

To solve the electronic structure problem, one of the most successful approximations comes in the form of density functional theory (DFT). Here a different approach is taken to the problem by grouping the electrons together in an object known as the electron-density. This electron-density is a function that only depends on the three spatial coordinates x, y, and z. In this way the degrees of freedom in the system are massively reduced. This also lowers the amount of required computing power. One of the problems however, is that the exact description of a quantum system, as a function

of the electron-density, still remains to be found. This is why, for the time being, we have to make due with educated guesses. Applications of DFT range all the way from the scientific world to industry. An example of an industrial application of DFT is that it aids in understanding the degradation of polymers through the calculation of bond dissociation energies [5].

Another research field that has been steadily gaining traction in the 21st century is machine learning. This field mainly concerns itself with the development of algorithms and mathematical techniques that help computers find and exploit patterns in data. These techniques range from simple regression, based on only a few parameters, to huge neural networks that contain millions of parameters. A great example of an application of these techniques is image recognition. If someone would ask you to write an algorithm that differentiates cats from dogs you would probably have no idea where to start, even though you are perfectly capable of doing the task yourself. Without the use of machine learning you would have to manually come up with some set of criteria that differentiates images of cats from those of dogs. With the aid of machine learning, combined with a large data set of labeled example images, one can easily "train" computers to carry out this task. This training procedure typically entails that the algorithm, *e.g.* a neural network (NN), is provided with a lot of examples along with the desired model output. The performance of the algorithm is then evaluated based on how well it is capable of reproducing these outputs, after which the parameters are changed with the hopes of improving the performance. This iterative procedure is referred to as the training procedure. In recent years machine learning has been applied to increasingly more complicated tasks such as the generation of moving footage of people, known as deepfakes, and voice cloning [6, 7]. The range of applications has also been extended to the speedup of physics simulations, such as the simulation of granular materials and fluids, which are typically quite computationally demanding [8].

Nowadays, machine learning approaches are also becoming widely applied to the field of quantum chemistry. Previous research has mainly focused on the direct prediction of a subset of molecular properties from molecular structures, such as ground state energies for the use in molecular dynamics simulations [9]. NNs can be quite useful for this since they are differentiable with respect to their input. Other approaches have also been proposed that utilize results from, computationally cheaper, lower level methods with the aim of using machine learning techniques to reproduce higher level, and thus more accurate, results [10, 11]. However, quite recently a new trend has started to emerge. In the DFT formalism, most of the interesting properties of a system can be derived from the electron-density. A promising approach would therefore be to directly predict this electron-density, as opposed to only a set of molecular properties. In [12]

it has been shown that NNs are capable of reproducing the electron-density of multi-electron systems in simple potentials. An approach that is more directly applicable to quantum chemistry has been proposed by [13], where instead of directly targeting the electron-density, they focused on the Fock matrix. This Fock matrix encodes the same information as the density, expressed in a finite basis set of atom-centred basis functions. After a single diagonalization one then obtains the predicted electron-density expressed in the considered basis. With this in mind, they attempted to predict these Fock matrices for a couple of small molecules. For this they trained separate NNs, each targeting a single molecule, on a large data set of different conformations. Subsequently they showed that such a model, trained specifically on conformations of a single molecule, is capable of interpolating between these data points, mostly within chemical accuracy.

In this thesis, we will introduce some novel ways of applying the NN machinery to quantum chemistry with the hopes of obtaining accurate result for a fraction of the computational cost, while retaining access to the electronic-structure. For this purpose we will focus on the Fock matrix as present in both DFT and density functional based tight binding (DFTB). As stated earlier, this approach gives us the benefit of obtaining the electron-density, and by this, the density dependent properties of the system. We will start off by treating only a single molecule, namely water, and try to construct the DFT Fock matrix from an initial guess based on the superposition of free-atom densities. Next, we shift our focus to DFTB which is a less accurate semi-empirical approximation of DFT. The aim here will be to construct an approach that generalizes well between different molecules. For this part we build upon the work of [14]. Finally, we attempt to achieve the same generalizability for DFT, again employing an initial guess based on free-atom densities. Before going into our methodologies and experiments, let us first consider the necessary theory.

Chapter 2

Theory: Quantum chemistry

From a physical point of view, a molecule can be pictured as a collection of positively charged nuclei and negatively charged electrons. In the non-relativistic limit the behaviour of these particles is accurately described by the Schrödinger equation [15]. However, for molecular systems, extracting the behaviour of these particles from this equation turns out to be quite problematic. This, because of the many-body nature of such systems, in addition to the apparent coupling between the nuclear and electronic degrees of freedom. In this chapter we will introduce how is typically dealt with these issues, allowing us to study increasingly more complex systems.

2.1 Molecular Hamiltonian

The most widely applied approximation in quantum chemistry is the Born-Oppenheimer (BO) approximation. In this approximation we consider the nuclei as stationary point particles surrounded by moving electrons. This is a well motivated approximation that is based on the fact that, in molecules, the mass of the atomic nuclei is generally three orders of magnitude larger than the mass of an electron. This means that in comparison to the movement of the nuclei, the motion of the electrons is almost instantaneous. In this context, the BO approximation allows us to uncouple the electronic and nucleic degrees of freedom.

Starting from this, the electronic Hamiltonian of a molecular system $\hat{H}^{mol}(\mathbf{R})$ can be written as an operator that explicitly depends on the positions $\mathbf{R}$ of the stationary

nuclei. This reads[1] [16]

$$\hat{H}^{mol}(\mathbf{R}) = \hat{T}_e + \hat{V}_{en}(\mathbf{R}) + \hat{V}_{ee} + E_{nn}(\mathbf{R}), \tag{2.1}$$

where

$$\hat{T}_e = -\frac{1}{2}\sum_{i=1}^{N} \nabla^2_{\mathbf{r}_i} \tag{2.2}$$

is the kinetic energy of the electrons,

$$\hat{V}_{en}(\mathbf{R}) = \sum_{i=1}^{N}\sum_{\mu=1}^{N_{nuc}} \frac{-Z_\mu}{|\mathbf{r}_i - \mathbf{R}_\mu|} \tag{2.3}$$

the Coulomb attraction between the nuclei and the electrons,

$$\hat{V}_{ee} = \sum_{1\leq i<j\leq N} \frac{1}{|\mathbf{r}_i - \mathbf{r}_j|} \tag{2.4}$$

the Coulomb repulsion between the electrons, and

$$E_{nn}(\mathbf{R}) = \sum_{1\leq \mu<\nu\leq N_{nuc}} \frac{Z_\mu Z_\nu}{|\mathbf{R}_\mu - \mathbf{R}_\nu|} \tag{2.5}$$

the Coulomb repulsion between the nuclei. Here, N is the number of electrons present in the system, $\mathbf{r}$ the position of the electrons, N_{nuc} the number of nuclei, and Z the corresponding charge of each nucleus. This Hamiltonian now entirely describes the BO approximated system. Determining the properties of the system requires solving for its eigenstates. The corresponding eigenvalue equation looks as follows:

$$\hat{H}^{mol}\,|\Psi_n\rangle = E_n\,|\Psi_n\rangle \tag{2.6}$$

with $|\Psi_n\rangle$ being the n-th eigenstate and E_n the corresponding energy. This equation is known as the time-independent Schrödinger equation [15]. Usually we are most interested in the ground state $|\Psi_0\rangle$ with corresponding energy E_0 [4, 17, 18].

2.2 Hartree-Fock

One of the first approaches to finding the ground state of the molecular Hamiltonian (equation 2.6) is known as Hartree-Fock (HF). This approach is based on the mean-field approximation where each electron interacts with the mean field generated by the other

[1] Note that we consider atomic units: $e = m_e = \hbar = 4\pi\epsilon_0 = 1$ such that distance is measured in Bohr $= a_0 = \frac{4\pi\varepsilon_0\hbar^2}{m_e e^2}$ and energy in Hartree $= \frac{m_e}{\hbar^2}(\frac{e^2}{4\pi\varepsilon_0})^2$.

electrons. To start off, we express the ground state as a single Slater determinant (SD) constructed from a set of N one-electron spin-orbitals $\{\phi_i(\mathbf{x})\}$, *i.e.*

$$|\Psi_0\rangle \rightarrow \psi^{SD}(\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N) = \frac{1}{\sqrt{N!}} \begin{vmatrix} \phi_1(\mathbf{x}_1) & \phi_2(\mathbf{x}_1) & \cdots & \phi_N(\mathbf{x}_1) \\ \phi_1(\mathbf{x}_2) & \phi_2(\mathbf{x}_2) & \cdots & \phi_N(\mathbf{x}_2) \\ \vdots & \vdots & \ddots & \vdots \\ \phi_1(\mathbf{x}_N) & \phi_2(\mathbf{x}_N) & \cdots & \phi_N(\mathbf{x}_N) \end{vmatrix}. \tag{2.7}$$

where $\phi_i(\mathbf{x})$ constitutes of a spatial part and a spin part $\{|\uparrow\rangle, |\downarrow\rangle\}$, represented by the combined space-spin coordinate $\mathbf{x}$. Expressing the ground state as a SD ensures that the wavefunction is anti-symmetric with respect to interchanging two electrons, *i.e.*

$$\Psi(\ldots, \mathbf{x}_i, \ldots, \mathbf{x}_j, \ldots) = -\Psi(\ldots, \mathbf{x}_j, \ldots, \mathbf{x}_i, \ldots). \tag{2.8}$$

This constraint stems from the Pauli-exclusion principle which states that no two electrons can simultaneously occupy the same quantum state. The energy of a single SD wavefunction is now obtained as

$$\begin{aligned} E^{SD}[\{\phi\}] &= \langle \psi^{SD} | \, \hat{H}^{mol} \, | \psi^{SD} \rangle \\ &= \sum_{i=1}^{N} \langle \phi_i | \hat{h}^{core} | \phi_i \rangle + \sum_{i<j}^{N} [\langle \phi_i \phi_j | \phi_i \phi_j \rangle - \langle \phi_i \phi_j | \phi_j \phi_i \rangle] + E_{nn}, \end{aligned} \tag{2.9}$$

where

$$\hat{h}^{core} = -\frac{1}{2} \nabla^2_{\mathbf{r}_i} + \sum_{\mu=1}^{N_{nuc}} \frac{-Z_\mu}{|\mathbf{r}_i - \mathbf{R}_\mu|} \tag{2.10}$$

is the single-electron operator and

$$\langle \phi_i \phi_j | \phi_k \phi_l \rangle = \int \int \phi_i^*(\mathbf{x}_1) \phi_j^*(\mathbf{x}_2) \frac{1}{|\mathbf{r}_1 - \mathbf{r}_2|} \phi_k(\mathbf{x}_1) \phi_l(\mathbf{x}_2) d\mathbf{x}_1 d\mathbf{x}_2. \tag{2.11}$$

In this last equation $\int d\mathbf{x}$ represents the integral over all three-dimensional space and spin.

To obtain the HF single SD wavefunction one has to resort to the variational principle. The variational principle states that any trial wavefunction has an energy that is equal or larger than the exact ground state energy. Therefore, the single SD solution is found by minimizing the energy (equation 2.9) with respect to the one-electron orbitals $\{\phi_i(\mathbf{x})\}$. It turns out that this is equivalent to solving the eigenvalue equation

$$\hat{F}[\{\phi\}] \, |\phi_i\rangle = \varepsilon_i \, |\phi_i\rangle \,, \tag{2.12}$$

known as the HF equations. The Fock-operator $\hat{F}$ explicitly depends on its eigenfunctions $\{|\phi_i\rangle\}$, denoted by $[\{\phi\}]$, with respective eigen energies $\{\epsilon_i\}$. It is defined as

$$\hat{F}[\{\phi\}] = \hat{h}^{core} + \hat{V}_H[\{\phi\}] + \hat{V}_x[\{\phi\}] \tag{2.13}$$

with

$$\hat{V}_H[\rho](\mathbf{x}) = \int \frac{\rho(\mathbf{x}')}{|\mathbf{r} - \mathbf{r}'|} d\mathbf{x}' \tag{2.14}$$

resembling the classic Coulomb potential, typically referred to as the Hartree term. In this definition we used $\rho(\mathbf{x})$ for the electron-density which is defined as

$$\rho(\mathbf{x}) = \sum_{i=1}^{N} |\phi_i(\mathbf{x})|^2. \tag{2.15}$$

There is also the exchange potential

$$\hat{V}_x[\{\phi\}](\mathbf{x}, \mathbf{x}') = -\sum_{i=1}^{N} \frac{\phi_i(\mathbf{x})\phi_i^*(\mathbf{x}')}{|\mathbf{r} - \mathbf{r}'|} \tag{2.16}$$

given as an integral kernel. This means that letting the operator $\hat{V}_x$ act on ϕ_j results in

$$(\hat{V}_x\phi_j)(\mathbf{x}) = -\sum_{i=1}^{N} \phi_i(\mathbf{x}) \int \frac{\phi_i^*(\mathbf{x}')}{|\mathbf{r} - \mathbf{r}'|} \phi_j(\mathbf{x}') d\mathbf{x}'. \tag{2.17}$$

To solve equation 2.12 for the eigenstates, the eigenfunctions can be expressed as a linear combination of M basis functions $\{\chi_i(\mathbf{x})\}$ weighted by the elements of the coefficient matrix $\mathbf{C}$, *i.e.*

$$\phi_i(\mathbf{x}) = \sum_{j=1}^{M} \mathrm{C}_{ji}\chi_j(\mathbf{x}). \tag{2.18}$$

These are often referred to as molecular orbitals (MOs). The discretized Fock matrix $\mathbf{F}$ and overlap matrix $\mathbf{S}$ are now obtained as

$$\begin{aligned} \mathrm{F}_{ij} &= \langle\chi_i|\hat{F}|\chi_j\rangle = \langle\chi_i|[\hat{h}^{core} + \hat{V}_H + \hat{V}_x]|\chi_j\rangle \\ &= \langle\chi_i|\hat{h}^{core}|\chi_j\rangle + \sum_{\mu,\nu}^{M} \gamma_{\nu\mu} \langle\chi_i\chi_\mu|\chi_j\chi_\nu\rangle - \sum_{\mu,\nu}^{M} \gamma_{\nu\mu} \langle\chi_i\chi_\mu|\chi_\nu\chi_j\rangle, \end{aligned} \tag{2.19}$$

with discretizations presented in the order of the operators, and

$$\mathrm{S}_{ij} = \langle\chi_i|\chi_j\rangle. \tag{2.20}$$

Here we used $\boldsymbol{\gamma}$ to represent the one-body reduced density matrix (1RDM) which explicitly depends on $\mathbf{C}$ as

$$\gamma_{ij} = \sum_{k=1}^{N} \mathrm{C}_{ik}\mathrm{C}_{kj}^{\dagger}. \tag{2.21}$$

The coefficient matrix is then found by solving the eigenvalue equation, known as the Roothaan-Hall equations,

$$\mathbf{F}[\boldsymbol{\gamma}]\mathbf{C} = \mathbf{S}\mathbf{C}\mathrm{diag}(\boldsymbol{\varepsilon}) \tag{2.22}$$

self-consistently [19]. This means that an initial guess for the 1RDM $\boldsymbol{\gamma}^0$ is required, after which

$$\mathbf{F}^{n+1}[\boldsymbol{\gamma}^n]\mathbf{C}^{n+1} = \mathbf{S}\mathbf{C}^{n+1}\mathrm{diag}(\boldsymbol{\varepsilon})^{n+1} \tag{2.23}$$

is repeatedly solved for $\mathbf{C}$ until the solution converges. Note that the MOs obtained from equation 2.23, with corresponding MO energies $\{\varepsilon_i\}$, form an orthogonal set. The MOs corresponding to the N lowest eigenvalues, or orbital energies, are now used to construct the single SD HF ground state wavefunction minimizing the HF energy [4, 17, 18].

However, it turns out that a single SD is not enough to describe the ground state, *i.e.* a more extensive mathematical object is required. To find the exact eigenstates of the molecular Hamiltonian, within a given basis, one can employ full configuration-interaction (CI) which expresses the eigenstates as a linear combination of all possible SDs. However, a major downside to this is that solving the required equations scales factorially with the size of the system. To combat this issue there are several post-HF methods available, such as truncated CI, coupled-cluster (CC), and Møller–Plesset perturbation theory (MP). These typically make use of the HF MOs to express the eigenstates in terms of a subset of the possible SDs. Full CI and post-HF methods are briefly discussed in appendix A [4].

2.3 Density functional theory

In the previous section, we have been looking for solutions for the ground state wavefunction of a given molecular system. It turns out that, most of the time, trying to solve the electronic structure problem with a high accuracy implies the derivation of a very complex wavefunction. In this context, the computational requirements of post-HF methods, *e.g.* Full CI, CC, *etc.*, increases dramatically with the size of the molecular system. To circumvent this problem, a very good alternative can be found in density functional theory (DFT).

In DFT, a new point of view is adopted in which one leaves the pure wavefunction theory to specifically focus on the electronic density. In this context, solving the electronic

ground state problem becomes practically easier: instead of determining a multi determinant wavefunction depending on many electronic coordinates, in DFT we focus on a single density function which depends on a unique coordinate $\mathbf{r}$ (the effective coordinate of a single electron). As a consequence, DFT has the advantage of being quite cheap computationally and represents one of the most prolific methods in quantum chemistry, especially when dealing with large bio-molecular systems.

DFT builds on the Hohenberg-Kohn theorems which state that the map between a given local potential $V(\mathbf{r})$ and the energy minimizing $|\Psi_0\rangle$ of the resulting Hamiltonian is invertible [20]. This means that, assuming $|\Psi_0\rangle$ is known, we could in theory find a unique $V(\mathbf{r})$ (unique up to a constant) that matches this ground state. The second statement is that the map between $|\Psi_0\rangle$ and the resulting electron-density ρ is also invertible. These two statements together mean that $V(\mathbf{r})$, unambiguously describing the system, can be written as a functional of ρ, *i.e.* $V[\rho]$. This is what motivated the introduction of the exchange-correlation potential $\hat{V}_{xc}[\rho]$ and exchange-correlation energy $\hat{E}_{xc}[\rho]$. The two are related such that

$$\hat{V}_{xc}[\rho] = \frac{\delta E_{xc}[\rho]}{\delta \rho}, \tag{2.24}$$

or in words: the exchange-correlation potential is defined as the functional derivative of the exchange-correlation energy E_{xc} with respect to ρ. This is where the energy is compensated for the lack of real kinetic-energy and electron-electron repulsion, depending on ρ. It is an attempt to mimic all the effects that are too difficult to calculate. A huge amount of functionals for V_{xc} exist based on different assumptions and approximations [21]. To do DFT calculations, one typically resorts to the the Kohn-Sham equations. These equations are derived in a similar manner as the HF equations, effectively assuming a single SD $|\Psi_0\rangle$, and take the form

$$\hat{F}^{KS}[\rho]\,|\phi_i\rangle = \varepsilon_i\,|\phi_i\rangle\,, \tag{2.25}$$

Here the Kohn-Sham operator $\hat{F}^{KS}$ is defined as

$$\hat{F}^{KS}[\rho] = \hat{h}^{core} + \hat{V}_H[\rho] + \hat{V}_{xc}[\rho]. \tag{2.26}$$

Similarly to HF, the Kohn-Sham equations can be discretized and solved self-consistently within a given basis $\{\chi_i(\mathbf{x})\}$, once again yielding a set of orthogonal MOs. Because of the fact that a single SD $|\Psi_0\rangle$ is assumed to derive these equations, we can reuse the discretizations of $\hat{h}^{core}$ and $\hat{V}_H$ as presented in section 2.2 (equation 2.19). The discretization of $\hat{V}_{xc}$, namely $\langle\chi_i|\hat{V}_{xc}|\chi_j\rangle$, is evaluated numerically since typically no analytical solution exist. The discretized version of $\hat{F}^{KS}$ is also referred to as the Fock

matrix, since the procedures are essentially the same for both HF and DFT. In Kohn-Sham DFT the energy is defined as

$$E^{DFT}[\rho] = \sum_{i=1}^{N} \langle \phi_i | \left[\hat{h}^{core} + \frac{1}{2}\hat{V}_H[\rho] \right] | \phi_i \rangle + E_{xc}[\rho] + E_{nn}, \tag{2.27}$$

where E_{nn} is defined as before (equation 2.5) and E_{xc} is evaluated numerically [4, 17, 18].

2.4 Density functional based tight binding

Starting from the fundamental concepts introduced in DFT, another method has been recently introduced, namely density functional based tight binding (DFTB) [22, 23]. From a theoretical point of view, DFTB can be seen as a simplified version of the DFT procedure where some clever approximations have been introduced in the model. Here we will briefly discuss the model and its assumptions.

For the treatment of DFTB we start off by introducing the tight binding approximation which entails that electrons at lower energy levels are considered as non-interacting with the environment outside of the atom. By this approximation DFTB only considers valence electrons. This means that instead of a system with N electrons we are left with N_{val} electrons. For the sake of simplicity we will stick to the spin-restricted formalism suitable for closed-shell systems, *i.e.* systems with an even number of electrons. This means that the $N_{val}/2$ MOs corresponding to the lowest eigenvalues are doubly occupied, accounting for spin.

In deriving the DFTB equations one starts off with ρ_0 which is the electron-density composed of free-atom densities, *i.e.* the atoms in the system are free and uncharged. These only have to be calculated once for each atom and for a given functional. Consequently, ρ_0 is expressed as a superposition of free-atom densities ρ_0^μ centered on their respective nucleus μ:

$$\rho_0 = \sum_{\mu=1}^{N_{nuc}} \rho_0^\mu, \tag{2.28}$$

where ρ_0 does not minimize $E[\rho]$, but neighbors the true minimizer ρ by a small fluctuation $\delta\rho$, *i.e.*

$$\rho = \rho_0 + \delta\rho. \tag{2.29}$$

The DFT energy (equation 2.27) is now expanded up to second order in $\delta\rho$ around ρ_0. The resulting energy expression is summarized in the following equation:

$$\begin{aligned} E^{DFTB} &= \sum_{i=1}^{N_{val}/2} 2\,\langle\phi_i|\,\hat{F}^0[\rho_0]\,|\phi_i\rangle + E_{rep}[\rho_0] + E_{coul}[\delta\rho,\rho_0] \\ &= E_{band}[\rho_0] + E_{rep}[\rho_0] + E_{coul}[\delta\rho,\rho_0], \end{aligned} \tag{2.30}$$

with

$$\hat{F}^0 = \hat{h}^{core} + \hat{V}_H[\rho_0] + \hat{V}_{xc}[\rho_0] \tag{2.31}$$

being the DFTB approximated Kohn-Sham operator. The first energy term E_{band} is the band-structure energy defined as the sum of occupied orbital energies. The other two terms are the repulsion energy E_{rep}, which contains the nuclear-nuclear repulsion term, and the Coulomb term E_{coul}, which contains the energy contribution from the distribution of electronic charges. Exact definitions of these terms as well as the entire derivation are presented in [22] and [23]. For DFTB the MOs are typically expressed as a linear combination of M valence orbital basis functions of Slater-type $\{\chi_i^\mu(\mathbf{r})\}$ centred on nucleus μ, *i.e.*

$$\phi_i(\mathbf{r}) = \sum_{\mu=1}^{N_{nuc}} \sum_{j\in\mu} \mathrm{C}_{ji}\chi_j^\mu(\mathbf{r}). \tag{2.32}$$

Accounting for the first order corrections in $\delta\rho$ only E_{band} and E_{rep} are included in the definition for the energy. This is referred to as DFTB1. In DFTB1 the MOs are obtained by solving the eigenvalue equation

$$\mathbf{F}^0\mathbf{C} = \mathbf{S}\mathbf{C}\mathrm{diag}(\boldsymbol{\varepsilon}), \tag{2.33}$$

where $\mathbf{S}$ is defined in the same way as for HF and DFT (equation 2.20). However, with regards to $\mathbf{F}^0$ one final approximation is done, namely

$$\mathrm{F}^0_{ij} = \begin{cases} \langle\chi_i^A|\hat{F}^0[\rho_0=\rho_0^A, \hat{V}_{en}=\hat{V}_{en}(\mathbf{R}_A)]|\chi_i^A\rangle \text{ if } i=j,\ i,j\in A \\ \langle\chi_i^A|\hat{F}^0[\rho_0=\rho_0^A+\rho_0^B, \hat{V}_{en}=\hat{V}_{en}(\mathbf{R}_A,\mathbf{R}_B)]|\chi_j^B\rangle \text{ if } i\in A,\ j\in B,\ A\neq B \\ 0 \text{ if } i\neq j,\ i,j\in A \end{cases}, \tag{2.34}$$

such that only density contributions and Coulomb attractions from the atoms considered in the matrix element are taken into account. Note that this equation does not need to be solved self-consistently since $\hat{F}^0$ depends solely on ρ_0. E_{rep} is now approximated as a pairwise potential $V^{rep}(R_{IJ})$ fitted to atomic forces between atom pairs in different

environments as obtained from accurate DFT calculations. It takes the following form:

$$E_{rep} = \sum_{i<j}^{N_{nuc}} V_{ij}^{rep}(R_{ij}), \tag{2.35}$$

with R_{ij} being the distance between nucleus i and j.

E_{coul} can be included in this procedure by going to DFTB2 or self-consistent charge (SCC)-DFTB. To do this one first expresses the net Mulliken charge [24], noted Δq, on a single atom A as

$$\Delta q_A = Z_A - \frac{1}{2} \sum_{i=1}^{N_{val}/2} \sum_{\mu \in A} \sum_{\nu=1}^{M} 2(\mathrm{C}_{\mu i}\mathrm{S}_{\mu\nu}\mathrm{C}_{\nu i} + \mathrm{C}_{\nu i}\mathrm{S}_{\nu\mu}\mathrm{C}_{\mu i}). \tag{2.36}$$

Using this definition the SCC-DFTB Fock matrix $\mathbf{F}^{SCC}$ is expressed as

$$\mathrm{F}_{ij}^{SCC} = \mathrm{F}_{ij}^{0} - \frac{1}{2}\mathrm{S}_{ij} \sum_{C=1}^{N_{nuc}} (\gamma_{AC} + \gamma_{BC})\Delta q_C \text{ with } i \in A, j \in B, \tag{2.37}$$

which does require a self-consistent procedure to diagonalize. E_{coul} is now approximated as

$$E_{coul} = \frac{1}{2} \sum_{A,B}^{N_{nuc}} \Delta q_A \gamma_{AB} \Delta q_B \tag{2.38}$$

with

$$\gamma_{AB} \approx \gamma_{AB}(U_A, U_B, R_{AB}) \tag{2.39}$$

depending on the Hubbard parameter U_i of both atoms. This parameter is defined as the second derivative of the free-atom energy E^{atom} with respect to the number of electrons N_e:

$$U_A = \frac{\partial^2 E^{atom}}{\partial N_e^2}. \tag{2.40}$$

Tabulated Hubbard parameters can be found in [25] and the definition for approximated γ_{AB} in [26].

Chapter 3

Theory: Neural networks

In this thesis we will make use of one of the most famous machine learning models, namely the neural network (NN). This model is much more versatile than standard statistical methods such as linear regression, logistic regression, and even nonlinear-regression. The main advantage comes from the fact that a NN is capable of recognising complex non-linear relationships in large higher-dimensional data sets. This, without the need of supplying an ansatz for the relation in question. This means that a NN can find relations that the human eye is not capable of detecting. This interesting property makes NNs promising tools to predict and extrapolate the non-linear relationships present in atomic interactions. Let us now go into how a NN works.

3.1 The model

In biological systems neurons are connected with each other to form complex networks with the goal of transmitting and processing information. An example of this would be the reception and processing of visual information in the brain. When visible light hits the retina it results in the activation of a set of neurons. Through connected neurons the signal is then propagated to the brain where the information is processed. Based on the resulting activated neurons in the brain it decides whether the visual stimulus corresponds to danger, food, a sexual partner *etc.*, and how to act from here. When the associated response results in a beneficial outcome, such as obtaining food, the responsible neural connections are made stronger, while for a non-beneficial outcome they are weakened. This essentially describes the learning process in a hugely simplified manner. Artificial NNs now constitute of a set of artificial neurons, or nodes, and their connections with the aim of mimicking their biological counterparts along with the associated learning process [27, 28].

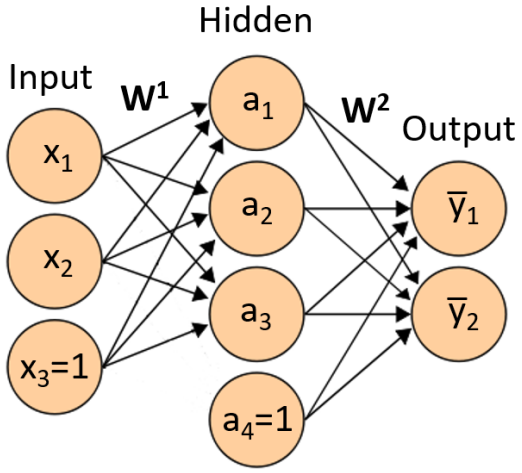

FIGURE 3.1: Schematic depiction of the different layers in a NN [1].

In this chapter we will focus on a single type of NN, namely a fully-connected feedforward NN. This type of network contains a single input layer of nodes where the "stimulus" is provided. This stimulus is then propagated to the next layer of nodes through the different connections. A schematic of this is shown in figure 3.1. Here we can see that each node of the input layer is connected to all of the nodes of the succeeding, hidden, layer. This is where the term "fully-connected" comes from. The term "feedforward" comes from the fact the information is propagated only in one direction. The information is finally passed on to the output layer which gives the "response" of the NN.

In figure 3.1 the initial stimulus of the input nodes is represented by $\mathbf{x} = (\mathrm{x}_1, \mathrm{x}_2, \mathrm{x}_3)^T$ which is a vector of numbers quantifying the activation of the nodes in the input layer. This input vector contains the independent variables on which we want to base our predictions. The NN now functions as a map to model the dependency of the output/dependent variables $\mathbf{y} = (y_1, y_2)^T$ on $\mathbf{x}$. The model predictions are contained in $\bar{\mathbf{y}} = (\bar{y}_1, \bar{y}_2)^T$. To obtain the activations $\mathbf{a}$ of the nodes in the hidden layer, starting from $\mathbf{x}$, one first does

$$\mathbf{z}^1 = \mathbf{W}^1\mathbf{x}, \tag{3.1}$$

where $\mathbf{W}^1$ is the matrix describing the connections between the input layer and the hidden layer. $\mathbf{W}^1$ is simply a matrix containing 3×3 parameters, known as weights, which scale the inputs. This means that $\mathbf{z}^1$ is simply a linear combination of the activations at the input nodes. To finally obtain $\mathbf{a}$, the elements of $\mathbf{z}^1$ are subjected to activation function $A(x)$ such that

$$\mathbf{a} = A(\mathbf{z}^1). \tag{3.2}$$

This activation function is responsible for the non-linear behaviour of the NN. The most used form for $A(x)$ is the rectified linear unit (ReLU):

$$A(x) = \begin{cases} x \text{ if } x \geq 0 \\ 0 \text{ if } x < 0 \end{cases} . \tag{3.3}$$

As we can see in figure 3.1, both the input layer and the hidden layer contain a single node where the activation is always set to one. These nodes are known as bias nodes and improve the versatility of the model. Therefore, after obtaining $\mathbf{a}$ from equation 3.2, a single one is added to this vector. The model response or prediction is then given by

$$\mathbf{z}^2 = \mathbf{W}^2\mathbf{a} \tag{3.4}$$

$$\bar{\mathbf{y}} = M(\mathbf{z}^2), \tag{3.5}$$

where $\mathbf{W}^2$ is the 4×2 matrix containing the weights mapping the hidden layer to the output layer. At this final step the activation function $M(x)$ is applied which can take different forms depending on the task at hand. For regression purposes one usually sticks to a linear activation function, *i.e.* $M(x) = x$.

In practice one is free to choose the size/dimension of the hidden layers, as well as the amount of hidden layers, resulting in fewer or more parameters. Adding more hidden layers or increasing the size of the hidden layers increases the versatility of the model. To describe a model with H hidden layers one starts off by doing

$$\mathbf{a}^0 = \mathbf{x}, \tag{3.6}$$

with included bias node, and iterating

$$\mathbf{z}^{n+1} = \mathbf{W}^{n+1}\mathbf{a}^n \tag{3.7}$$

$$\mathbf{a}^{n+1} = A(\mathbf{z}^{n+1}) \tag{3.8}$$

while $n < H$. Note that a single one is added to $\mathbf{a}^{n+1}$ to account for the bias node in each layer. Finally, the prediction becomes

$$\mathbf{z}^{H+1} = \mathbf{W}^{H+1}\mathbf{a}^H \tag{3.9}$$

$$\bar{\mathbf{y}} = M(\mathbf{z}^{H+1}). \tag{3.10}$$

The size of each of the weight matrices $\{\mathbf{W}^i\}$ is determined by the number of nodes in the preceding (# columns) and succeeding (# rows) layers.

3.2 Training procedure

For the NN to make valid predictions, the weights $\{\mathbf{W}^i\}$ present in the model require optimization. This requires training data from which the model "learns". This training data can be represented as a set of J input vectors $\{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_J\}$ with corresponding observed outputs $\{\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_J\}$. For regression problems the difference between the model outputs and the observed/desired outputs can now be evaluated using the mean squared error (MSE). The goal is now to optimize $\{\mathbf{W}^i\}$ such that we minimize the MSE. This is equivalent to minimizing the sum of squared errors defined as

$$L(\mathbf{W}^1, \ldots, \mathbf{W}^{H+1}) = \sum_{i=1}^{J} ||\mathbf{y}_i - \bar{\mathbf{y}}(\mathbf{x}_i)||^2. \tag{3.11}$$

Such a function, which is used to evaluate the performance of the NN, is known as a loss function.

The easiest way to proceed now is to use gradient descent to optimize the parameters with respect to this loss function. This procedure is often employed to minimize some function $f(x)$ with respect to a parameter or variable x, and only requires the first derivative of the function $\frac{df}{dx}$. The value of the parameter is then updated in the direction of the gradient

$$x^{n+1} = x^n - \alpha \frac{df(x^n)}{dx}, \tag{3.12}$$

where $\alpha > 0$ is some parameter scaling the derivative. This step is then repeated until convergence or until n passes a certain threshold value. To apply this procedure to a NN the partial derivatives of the weights with respect to L are required. After obtaining all the partial derivatives for each of the weights, they can be updated as

$$\mathrm{W}_{ij} \leftarrow \mathrm{W}_{ij} - \alpha \frac{\partial L}{\partial \mathrm{W}_{ij}}. \tag{3.13}$$

Obtaining these partial derivatives is known as "backpropagation". This procedure makes extensive use of the chain-rule for differentiation. The first thing to do is find the gradient with respect to a single weight W_{ij}^{H+1} in $\mathbf{W}^{H+1}$, that maps the last hidden layer to the output layer. This gives

$$\frac{\partial L}{\partial \mathrm{W}_{ij}^{H+1}} = \frac{\partial L}{\partial \bar{\mathrm{y}}_i} \frac{\partial \bar{\mathrm{y}}_i}{\partial \mathrm{z}_i^{H+1}} \frac{\partial \mathrm{z}_i^{H+1}}{\partial \mathrm{W}_{ij}^{H+1}} \tag{3.14}$$

which is a rather simple expression. Going back one more layer, *i.e.* to $\mathbf{W}^H$, one has to be more careful. This, because a single weight affects only the activation of a single node in the layer with the same index. However, this weight indirectly affects the activation

of all the nodes in the succeeding layers. This means that differentiation with respect to all these different dependencies is required. Therefore, for W_{jk}^{H} one obtains

$$\frac{\partial L}{\partial \mathrm{W}_{jk}^{H}} = \sum_{i} \frac{\partial L}{\partial \bar{\mathrm{y}}_i} \frac{\partial \bar{\mathrm{y}}_i}{\partial \mathrm{z}_i^{H+1}} \frac{\partial \mathrm{z}_i^{H+1}}{\partial \mathrm{a}_j^{H}} \frac{\partial \mathrm{a}_j^{H}}{\partial \mathrm{z}_j^{H}} \frac{\partial \mathrm{z}_j^{H}}{\partial \mathrm{W}_{jk}^{H}}. \tag{3.15}$$

This reasoning can easily be extended to account for the partial derivatives with respect to deeper lying weights. Note that the derivatives with respect to the activation at a bias node are always zero. All of these derivatives can simply be obtained analytically and mostly depend on the activation of a subset of nodes for a given training sample. With regards to the ReLU activation function the discontinuity at $x = 0$ is ignored such that its derivative can be defined as [29]

$$\frac{dA}{dx} = \begin{cases} 1 \text{ if } x > 0 \\ 0 \text{ if } x \leq 0 \end{cases}. \tag{3.16}$$

The whole procedure can now be summarized as iterating:

- Feeding each data point into the NN and obtaining the resulting activation at each node (forward propagation).
- Obtaining the partial derivatives with respect to the loss function through back-propagation.
- Updating the parameters using gradient-descent.

To ensure stability, each of the independent variables is normalized to lie within $[0, 1]$ [30]. In order to speed up the convergence of the training procedure, the gradient is typically not evaluated on the entire training set, but only on a subset before updating. This is known as mini-batch gradient-descent [31]. Nowadays many different methods are available for optimizing the parameters [32]. In this thesis we will stick to the Adam optimization algorithm, described in [33] and [34].

The parameters describing the architecture of a NN, *i.e* the number of hidden layers and nodes present in a neural network, and the training procedure, *e.g* the scaling of the gradient or the mini-batch size, are typically referred to as "hyper-parameters". Optimizing these parameters with respect to the model performance is known as hyper-parameter tuning. This is usually a time consuming undertaking, since it typically requires training a multitude of different NNs using a different set of hyper-parameters for each of them [35].

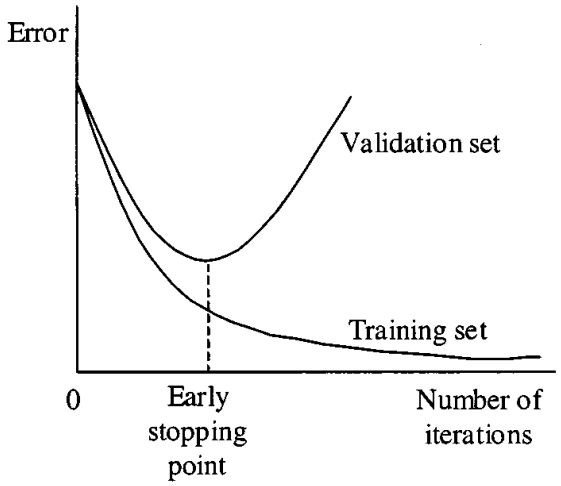

FIGURE 3.2: Overfitting during the training procedure and early-stopping [2].

3.3 Overfitting

An important issue that one has to be aware of during the training procedure is "overfitting". Overfitting means that the model becomes too specialized on the training data, such that its general performance goes down. This is a negative side-effect of working with such flexible models. To quantify overfitting, typically the help of a validation set is employed. This validation set contains a set of data points that are not used to optimize the model. One can keep track of the performance of the model on this validation set to spot any overfitting (figure 3.2). The easiest way of dealing with overfitting is to stop the training procedure when the validation set performance starts to go down. This is known as "early-stopping" and requires a patience parameter that determines for how many more iterations the training procedure continues after a minimum in validation set error is observed. The considered set of weights that results in the best performance on the validation set is then used for the final model [36]. Other methods to reduce overfitting include weight regularization [37], which places a constraint on the magnitude of the weights, and dropout regularization [38], which randomly removes some nodes from the network during the training procedure.

Chapter 4

Fock matrix in DFT

As stated earlier, the majority of this is thesis is aimed at applying the NN machinery directly to the DFT Fock matrix such that we obtain the electron-density and its derived properties. It is therefore useful to have a more in depth discussion about this object and what is represented by its matrix elements.

4.1 Matrix elements

The exact definition of the Fock matrix elements for DFT, analogous to HF, is given by

$$\mathrm{F}_{ij}^{DFT} = \langle \chi_i | \hat{F}^{KS}[\rho] | \chi_j \rangle \,, \tag{4.1}$$

where the operator itself only depends on the density. In addition to this density, the elements also depends on the chosen set of basis functions $\{\chi_i\}$. These basis functions are based on the exact electronic eigenfunctions of a single hydrogen atom (figure 4.1). They are real scalar functions depending on $\mathbf{r}$ centered on the atomic nucleus. The main two options for carrying out electronic structure calculations are Gaussian-type orbitals (GTOs) and Slater-type orbitals (STOs). A brief discussion about GTOs and STOs is presented in appendix B. Regarding the DFT part of this thesis we will employ a minimal Gaussian-type basis set, namely STO-6G, to reduce the amount of required predictions. We will also stick to the closed-shell formalism, *i.e.* we occupy each MO twice. This eliminates the need for spin-orbitals such that we only require spatial basis functions $\{\chi_i(\mathbf{r})\}$. Furthermore, we resort to the PBE functional for our DFT calculations and predictions [39, 40].

Let us now look at the converged Fock matrix for a single water molecule in STO-6G basis (figure 4.2). We can see here that the basis set contains the two $1s$ orbitals

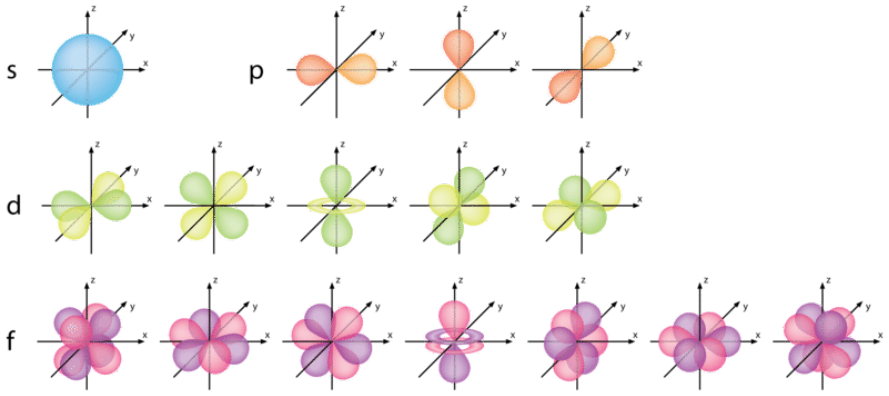

FIGURE 4.1: Exact atomic orbitals of a single hydrogen grouped by increasing angular momentum $s \to p \to d \to f$ [3].

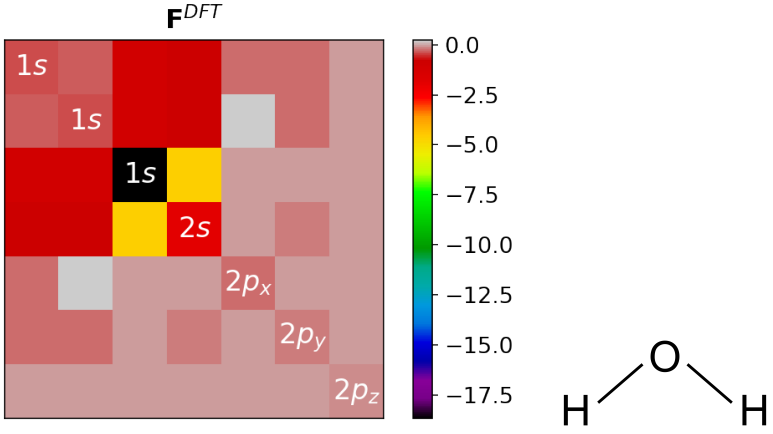

FIGURE 4.2: DFT Fock matrix for a single water molecule in STO-6G basis. Basis functions are labeled on their corresponding diagonal element.

from both hydrogen atoms, as well as the $1s$, $2s$, and the three p orbitals for oxygen. The elements of $\mathbf{F}^{DFT}$ (equation 4.1) now depend on the shape of two basis functions, the distance between their corresponding atoms, and the electron-density around them. Knowing the molecular structure and the basis functions we can already obtain $\mathbf{S}$, as well as the one-electron part of $\mathbf{F}^{DFT}$. The only purpose of the self-consistent field (SCF) procedure (section 2.2) is therefore to obtain the energy minimizing electron-density.

One important thing to realize is that the Fock matrix and the overlap matrix are not invariant to rotations of the molecule. We can easily see this if we take a look at the p orbitals in figure 4.1. For this we imagine an oxygen atom centred at the origin with its p-type basis functions oriented along the x, y, and z axis, just as in the figure. Now imagine a hydrogen atom located at distance R from the origin with a single $1s$-type basis function. In this situation the overlap between each of the p orbitals and the $1s$ basis function of the hydrogen depends on the exact orientation of $\mathbf{R}$ and not only on the radial distance. As a result the corresponding matrix elements of $\mathbf{F}^{DFT}$ and $\mathbf{S}$ will also rotate depending on $\mathbf{R}$. The rotation of molecules does not form an issue for DFT

since the final results will just rotate along with the molecule. However, any model that works with the Fock matrix as an input or output must also be able to deal with these rotations.

4.2 Initial guess based on the superposition of free-atom densities

In section 2.2 we have briefly discussed the workings of the SCF procedure. However, we have not yet discussed how this procedure is initialized. For this one first has to define the set basis set. This only requires knowledge about the molecular structure, including the type of atoms *e.g.* H, C, N, O *etc.* and the positions of the nuclei in space $\mathbf{R} = (\mathrm{R}_x, \mathrm{R}_y, \mathrm{R}_z)$, along with the type basis set one wishes to employ. After deciding upon the set of basis functions, *i.e.* their type (1s,2s,2p *etc.*) and centers, the overlap matrix and the one-electron part of the Fock matrix can be constructed. All that is left to do now is to obtain an initial guess for the electron-density. For this first guess there are several options available [41]. In this case we will stick to an initial guess based on the superposition of free-atom densities. This means that we start off by obtaining converged DFT densities for the individual atoms in the chosen basis set. A super position of these densities is then used to construct the initial 1RDM [41]. The first guess for the Fock matrix can now be constructed as

$$\mathrm{F}_{ij}^{guess} = \langle \chi_i | \hat{F}^{KS}[\boldsymbol{\gamma}^{init}] | \chi_j \rangle \,, \tag{4.2}$$

where $\boldsymbol{\gamma}^{init}$ is the 1RDM corresponding to the superposition of free-atom densities and $\mathbf{F}^{guess}$ the resulting Fock matrix. Our aim is now to use $\mathbf{F}^{guess}$ as a consistent starting point from which we can predict $\mathbf{F}^{DFT}$ using NN based models. This would effectively allow us to circumvent the SCF procedure such that only a single diagonalization is required to obtain the electron-density.

Chapter 5

Fock matrix corrections for water

In order to check the feasibility of such an approach, let us first focus on different conformations of a single molecule and try to see if we can accurately predict the converged Fock matrix from an initial guess based on free-atom densities. As stated earlier, this would eliminate the need for the SCF procedure. For this purpose we consider water as a model system.

5.1 SCF procedure

To see how the SCF procedure changes the electron-density of water we can take a look at figure 5.1. On the left we can see the free-atom based starting density $\rho^{init}(\boldsymbol{\gamma}^{init})$. What we observe is that the this initial electron-density is spread throughout the molecule. On the right we have the converged density, which is considerably more localized on the O atom. This is in line with the electric dipole present in a water molecule. This converged density can in turn be used to determine the properties of the system.

5.2 Method

To predict the converged Fock matrix of a water molecule, and subsequently the converged electron-density, one could make use of a simplified representation of the geometry. Such a representation includes the H-O-H angle θ and the two O-H bond lengths l_1 and l_2. However, the aim is to find a procedure which can easily be generalized to different systems. Consequently, we will not make use of this simplified description of the system in the construction of our model. Instead we propose the following approach.

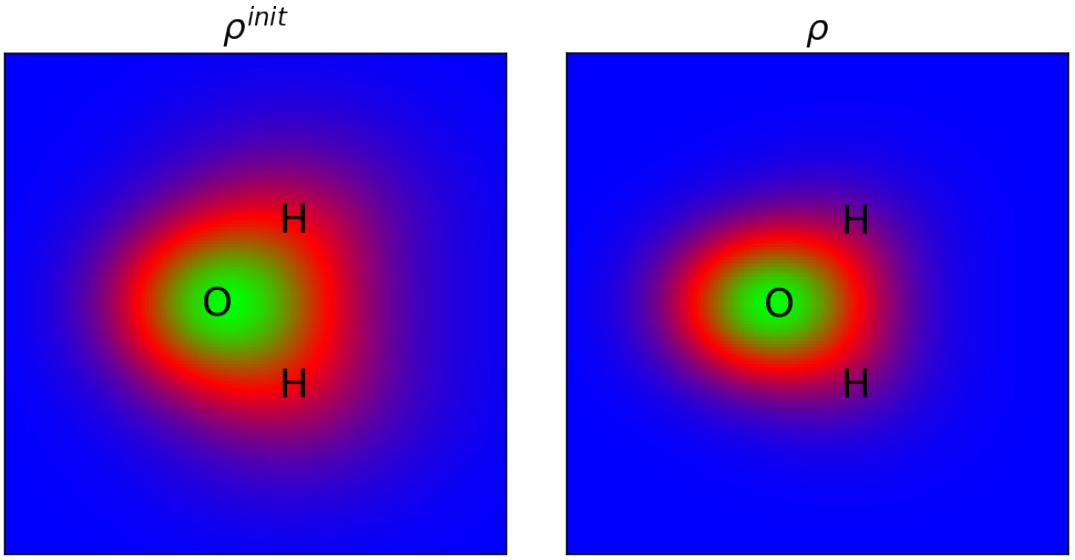

FIGURE 5.1: Superposition of free-atom electron densities for water in the $xy-$plane (left). Converged electron-density for water in the $xy-$plane (right). Density value from small to large: *blue* $\rightarrow$ *red* $\rightarrow$ *green*.

5.2.1 Model description

As stated earlier we are going to start our predictions from an initial guess based on free-atom densities. After obtaining this initial density we build our first guess for the Fock matrix $\mathbf{F}^{guess}$ expressed in the minimal basis set STO-6G. This is the starting point for our model. Next, we define our model targets $\mathbf{Y}$ as the matrix containing the desired corrections

$$\mathrm{Y}_{ij} = \mathrm{F}_{ij}^{DFT} - \mathrm{F}_{ij}^{guess}. \tag{5.1}$$

As a model we are going to work with a single fully-connected NN for which we define the following inputs:

- The first $5 \times K$ input nodes are reserved for the environment of a single atom. Here K is the number of nearest neighbours taken into account. In the case of water we only require $K = 2$. For each neighbor we supply five parameters, namely: the charge Z, the distance r with respect to the considered atom, and the three cartesian components of the normalized distance vector $\hat{\mathbf{r}}$ (*i.e.* $\hat{\mathrm{r}}_x, \hat{\mathrm{r}}_y$ and $\hat{\mathrm{r}}_z$). This information is supplied for each neighbor in the order of their distance.

- We reserve K input nodes for describing which interaction we are dealing with. For this we use one-hot encoding where $(1, 0, \ldots)$ is the nearest neighbor, $(0, 1, \ldots)$ the second nearest neighbor *etc.* [42]. When dealing with elements corresponding to basis functions centred on the same atom we only supply zeros here.

In both input and output layers, we allocate nodes to the elements of $\mathbf{F}^{guess}$ and $\mathbf{Y}$, respectively:

- A single node for the H-1s diagonal element.
- 15 nodes for the upper triangle of the O-block.
- A single node for the [H-1s]-[H-1s] interaction element.
- 5 nodes for the H-O interaction elements.

Each input vector contains only the relevant information for the prediction in question. The remaining input nodes are set to zero. During the training procedure the unused output nodes are not taken into account when evaluating the gradient. In this way, for a single conformation of water, we end up with: two input vectors for H, a single for O, two for the H-O interactions, and two for the H-H interaction. Note that the H-H interaction is accounted for twice since each H can both be the considered or the neighboring atom. After feeding the information into the model we obtain the predicted correction matrix $\bar{\mathbf{Y}}$. As a final step we symmetrize this matrix, *i.e.*

$$\bar{\mathbf{Y}} \leftarrow \frac{\bar{\mathbf{Y}} + \bar{\mathbf{Y}}^T}{2}, \tag{5.2}$$

to account for double predictions, such as H-H in the case of water. After this we obtain the predicted Fock matrix $\mathbf{F}^{NN}$ as

$$\mathbf{F}^{NN} = \mathbf{F}^{guess} + \bar{\mathbf{Y}}. \tag{5.3}$$

5.3 Experiments & results

In this section we will take a look at the results for the proposed model. For this purpose we will carry out two experiments. In the first experiment we try to conserve the orientation of the water molecule as much as possible while changing the conformation along its three degrees of freedom and examine how well a NN is capable of doing the required corrections. In the second experiment we randomly rotate the molecule for each conformation and see how this affects the performance of the NN. For the experiments conducted in this section we use **PySCF** [43] for the electronic structure calculations and **Keras** [44] for the NN implementation. Unless stated otherwise we stick to atomic units for the presentation of our results [16].

5.3.1 Data set

For the first experiment we minimize the variation in orientation by putting the O atom on the origin and the first H atom on the positive x-axis. The second H is now constrained to the xy-plane with $y \geq 0$. We now construct the training set and test set as follows:

- Training set: $15 \times 15 \times 15$ conformations varying l_1, l_2, and θ on an evenly spaced grid with $0.8 \leq l_1, l_2 \leq 1.25$ Å and $80° \leq \theta \leq 125°$.
- Test set: $20 \times 20 \times 20$ conformations varying l_1, l_2, and θ on an evenly spaced grid with $0.5 \leq l_1, l_2 \leq 1.5$ Å and $50° \leq \theta \leq 150°$.

For the training procedure we randomly take away 30% of the conformations from the training set and from this construct the validation set. For the second experiment we randomly rotate the molecule in space before constructing $\mathbf{F}^{guess}$ and $\mathbf{F}^{DFT}$. For the training set the grid resolution is now decreased to $7 \times 7 \times 7$, but enhanced with 20 randomly rotated samples per grid point. The test set resolution remains the same with only a single random orientation per grid point. We again use 30% of the training data to construct the validation set. As described in section 3.2, we normalize our inputs to lie within the interval [0,1], based on their range in the training set.

5.3.2 Model parameters & hyper-parameter tuning

To get a feel for the complexity of the problem we need to evaluate different sets of hyper-parameters to see what gives the best results. For each training run we employ early-stopping to prevent overfitting. For this procedure we set the patience parameter to 500 iterations and let the model train for a maximum of 1e5 iterations to ensure convergence. Mini-batches are used to speed up the convergence. The mini-batch size is set to 500 samples. We employ the Adam optimization algorithm to optimize the weights and the ReLU activation function for the hidden layers. For the final activation we make use of a linear activation function to end up with the final predictions. During the training procedure the performance is evaluated according to the MSE. For the learning rate we consider $[0.005, 0.001, 0.0005]$. With regards to the remaining Adam parameters we stick to the default values proposed by the creators [33]. For the first experiment, with a set orientation, we observed that 0.005 performed the worst, we therefore disregard it for the second experiment. The remaining validation set errors for the set orientation are presented in figure 5.2. The results for the second experiment, concerning the random orientation, are depicted in figure 5.3. Based on these results

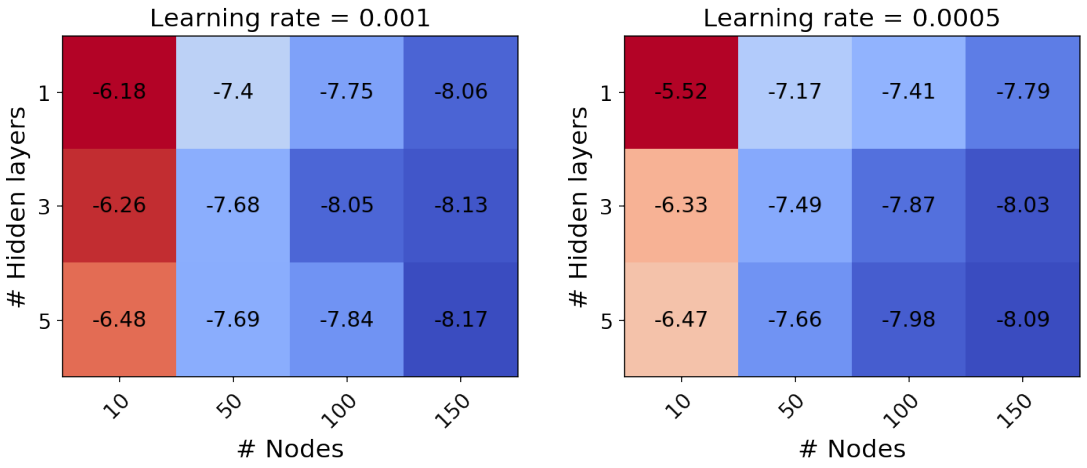

FIGURE 5.2: $\log_{10}$(MSE) for each of the hyper-parameter settings as evaluated on the validation set for water with a set orientation.

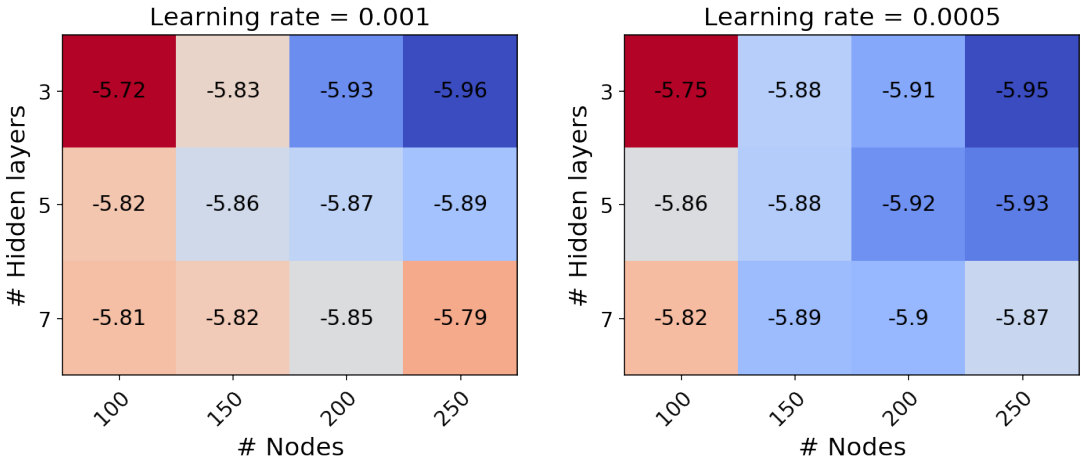

FIGURE 5.3: $\log_{10}$(MSE) for each of the the hyper-parameter settings as evaluated on the validation set for water with a random orientation.

we decide to go with three hidden layers for both data sets, with 150 nodes for the set orientation and 250 nodes for the random orientation. In both cases the learning rate is set to 0.001.

5.3.3 Model evaluation

Now that we have determined the set of hyper-parameters, we are interested in the performance of the model. For this we compare the mean absolute error (MAE) evaluated on the validation and the test set for both experiments (table 5.1). Here we find that in both cases the training and validation set error is almost the same. This means the model is capable of interpolating the points in the training data. The lack of rotation also seems to make it easier for a NN to do the corrections. The performance on the

TABLE 5.1: MAE for the two models evaluated on their respective, training, validation, and test sets.

	Set orientation		**Random orientation**	
	Guess	**NN**	**Guess**	**NN**
Training set	3.05e-2	1.30e-4	2.58e-2	6.11e-4
Validation set	3.06e-2	1.31e-4	2.59e-2	6.11e-4
Test set	3.67e-2	5.33e-3	3.09e-2	6.03e-3

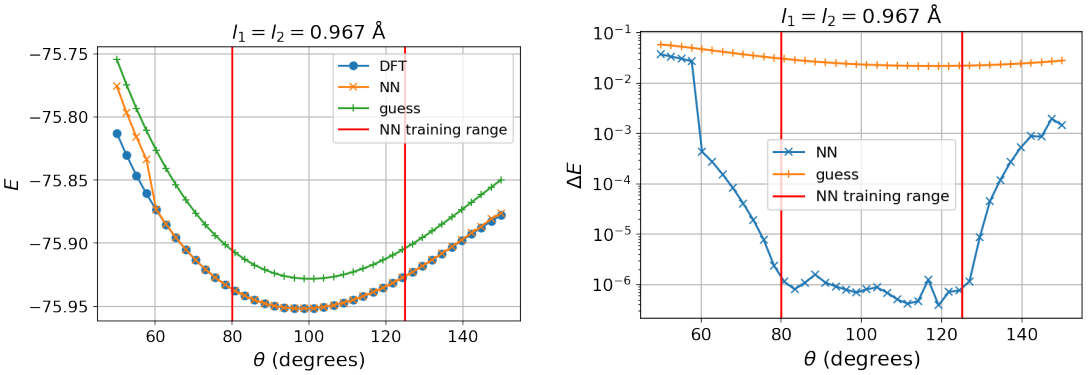

FIGURE 5.4: E (left) and ΔE (right) from the NN and guess, obtained after a single diagonalization, along the PES varying θ for water with a set orientation.

test set is however quite similar for both data sets, indicating limited extrapolation capabilities.

5.3.4 Results: Set orientation

To make a reasonable evaluation of the model we resort to examining the derived properties from the corrected Fock matrices. For this we consider the ground state energy E along a variation of the degrees of freedom present in the system, also known as a potential energy surface (PES). We also consider the energy difference with respect to the converged result ΔE. Here we treat the performance for a set orientation.

Let us first look at varying θ keeping both l_1 and l_2 stationary (figure 5.4). We can see that the NN performs well within chemical accuracy of $\sim 10^{-3}$ Hartree [13] along the PES, only exceeding this at the outside edges. We also note that the guess is improved upon along the entire PES. To get a global picture for the performance along all degrees of freedom we constructed figure 5.5 from the test set conformations. Here we can see that chemical accuracy is achieved well outside the training region, however the model seems to struggle when the atoms get close together. An improvement over the guess is achieved across almost the entire test set.

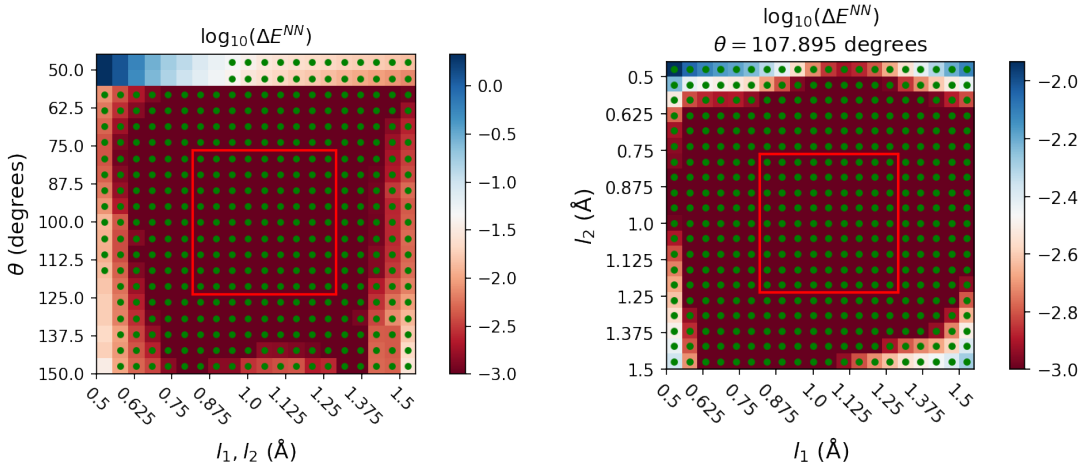

FIGURE 5.5: ΔE evaluated on both 2-dimensional PESs present in the test set for water with a set orientation. The training range of the NN is indicated by the red square. Green dots indicate improvement over the guess. The lower bound of the color bar is set to 10^{-3} Hartree to track chemical accuracy.

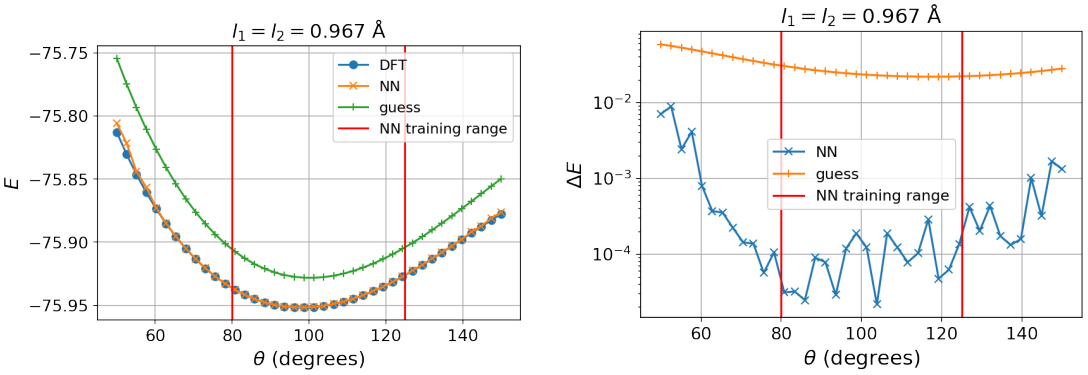

FIGURE 5.6: E (left) and ΔE (right) from the NN and guess, obtained after a single diagonalization, along the PES varying θ. Conformations were rotated randomly before corrections.

5.3.5 Results: Random orientation

Similar figures can be constructed for the data set with random orientations (figures 5.6 and 5.7). We again observe that the model improves upon the initial guess along the entire 1-dimensional PES. It even performs slightly better at smaller angles as compared to the other data set. However, within the training region the model performs significantly worse by about two orders of magnitude, while still remaining within chemical accuracy. We also observe larger fluctuations in ΔE within the training region. For the 2-dimensional PESs we observe a smaller region where the results remain within chemical accuracy, although within the training region this is not an issue.

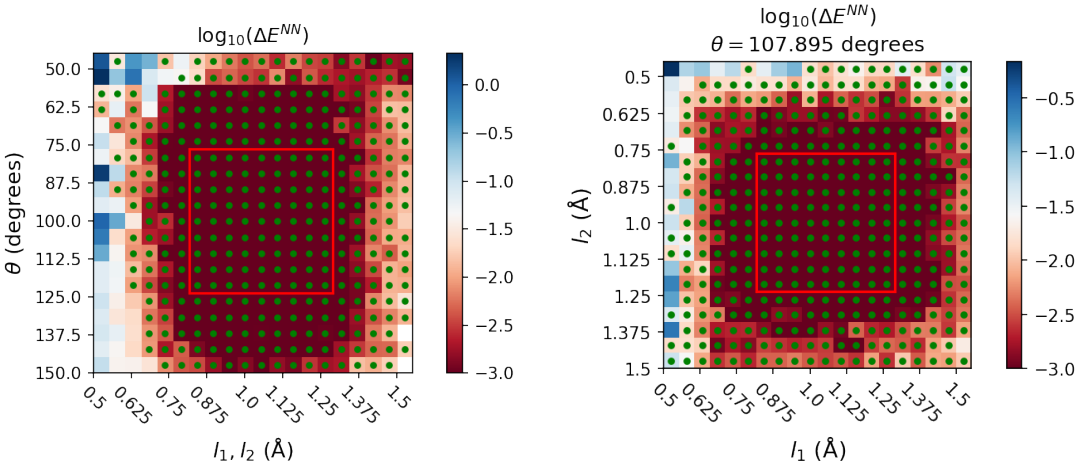

FIGURE 5.7: ΔE evaluated on both 2-dimensional PESs present in the test set for water with a random orientation. The training range of the NN is indicated by the red square. Green dots indicate improvement over the guess. The lower bound of the color bar is set to 10^{-3} Hartree to track chemical accuracy.

5.4 Discussion

In this chapter we defined our first model comprised of a single NN, focusing only on a water molecule. Here we explicitly refrained from using the simplified representation of the molecular geometry as to explore the possibilities of extending the model to other/larger molecules. For this purpose we carefully constructed a NN in which we minimize the reuse of the same input and output node for different descriptors. *E.g* the input node for the H-1s diagonal element of the guess matrix is different from that of O-1s. The same is true for the output nodes supplying the corrections. During the early stages of the model development this specialization of nodes helped us to reduce the error to chemically acceptable levels.

After deciding on the model description we trained and optimized the model on two data sets, one containing water in a set orientation and one containing random orientations of water. From the difference in both validation set error and resulting ground state energies we can conclude that these different orientations/rotations form quite the challenge for a NN. However, we also have to consider the buildup of the training and validation data set for both experiments. The combined training and validation portion of the first data set contained a higher resolution ($15 \times 15 \times 15$ versus $7 \times 7 \times 7$), while for the second data set this was enhanced with 20 random orientations. This means that for the first data set the combined training and validation set contained in total 3375 conformations, in contrast to 6820 rotations/conformations for the second data set. It is therefore difficult to compare these experiments one to one, since the performance on the second data set depends both on the resolution of the data and the amount rotations present. In both cases, a more extensive hyper-parameter tuning procedure might also improve the results. During the construction of the employed data sets guess and DFT results for all the random orientations were calculated independently. In the future it would be wise to use Wigner rotation matrices which are capable of rotating the Fock and overlap matrix along with the molecule [45]. This eliminates the need for effectively repeating the same DFT calculation.

However, we did find that a single NN is capable of reproducing the ground state energies of a water molecule through correcting the Fock matrix. Based on this, we propose an alternative to the approach presented in [13], where we now, instead of constructing the Fock matrix of a single molecule from scratch, exploit the power of a cheap and widely employed initial guess [41]. A good starting point for future research might be to try and combine the two methods.

Chapter 6

DFTB charge corrections

This chapter builds upon the work of high school students that conducted their final project at Software for Chemistry & Materials BV [14]. Here we shift our focus to DFTB and in particular the map between DFTB1 and SCC-DFTB. We will focus on modelling this map and try to make general predictions for small organic molecules. For this purpose we make use of a big data set of moleculer structures. Let us now first look at our approach.

6.1 Method

Similarly to DFT all the relavent information in DFTB is encoded in the overlap matrix and the Fock matrix. As described in section 2.4 we define these matrices in the basis of valence orbitals of Slater-type. Since the basis is unchanged between DFTB1 and SCC-DFTB the overlap matrix remains the same. Looking at the definitions for the Fock matrix in DFTB1 (equation 2.34) and SCC-DFTB (equation 2.37) we only require the net Mulliken charges Δq (equation 2.36) and the known Hubbard parameters to map one onto the other. Since these charges are obtained self-consistently, knowing these charges and diagonalizing the resulting Fock matrix is enough to bypass the self-consistent procedure. This means that we do not have to correct each Fock matrix entry separately, but only a single charge per atom. This massively reduces the amount of variables to predict. Another upside to this is that these charges are rotationally invariant.

6.1.1 Representing the atomic environment

In the previous chapter we defined the environment of a single atom on the basis of the nuclear charge, distance, and the normed distance vector of the neighbours. This way of representing the environment is however not invariant under rotations of the molecule. This means that the NN has to deal with these rotations and could likely give different predictions for the same molecular structures. One solution for this might be to supply many different rotations for each molecule in the training data.

However, a different approach has been pioneered by J. Behler [46]. He suggests the use of rationally invariant symmetry functions to define the environment of each atom. These symmetry functions can be divided into radial functions, constructed from two-body terms, and angular functions, constructed from three-body terms. Let us begin by providing the definition of the cut-off function f_c:

$$f_c(R_{ij}) = \begin{cases} 0.5[\cos(\frac{\pi R_{ij}}{R_c}) + 1] \text{ for } R_{ij} \leq R_c \\ 0 \text{ for } R_{ij} > R_c \end{cases}, \tag{6.1}$$

where R_c is the cut-off radius beyond which environmental information is not included. In our case we will stick to the following radial symmetry function describing the environment of atom i:

$$D_i^A = \sum_{j \in A} f_c(R_{ij}) e^{-\eta(R_{ij} - R_s)^2}, \tag{6.2}$$

where A represents all neighboring atoms of a single element type such that for every element we obtain a different D_i^A. Here η and R_s are parameters that we are free to choose. With regards to the three-body angular symmetry functions, we will stick to T defined as

$$T_i^{A,B} = 2^{1-\zeta} \sum_{j \in A} \sum_{\substack{k \neq j, \\ k \in B}} f_c(R_{ij}) f_c(R_{ik}) f_c(R_{jk}) (1 + \lambda \cos\theta_{ijk})^{\zeta} e^{-\omega(R_{ij}^2 + R_{ik}^2 + R_{jk}^2)} \tag{6.3}$$

describing the environment of atom i. Here A and B are again the sets of neighbouring atoms of single element type A or B. We also have tuneable parameters ω, ζ, and λ, as well as θ_{ijk} being the angle between atoms i, j, and k given by

$$\theta_{ijk} = \arccos(\frac{\mathbf{R}_{ij} \cdot \mathbf{R}_{ik}}{R_{ij} R_{ik}}). \tag{6.4}$$

Multiple sets of parameter values are typically used simultaneously to sufficiently explore the environment.

6.1.2 Model description

In this case we consider two different single NN models both containing a single output node for the charge correction

$$y = \Delta q^{SCC} - \Delta q^{DFTB1} \tag{6.5}$$

on each atom, where Δq^{SCC} and Δq^{DFTB1} are the net Mulliken charges obtained from SCC-DFTB and DFTB1, respectively. For both models we supply the atom-type as a one-hot encoded vector along with the DFTB1 charge. The difference now comes from how we define the environment. In the first model we take the previous approach (section 5.2.1) and supply Z, r, $\hat{\mathbf{r}}$, and the DFTB1 charge of the K nearest neighbours ordered by increasing r. Since we are now dealing with more than two element types, we supply Z as a one-hot encoded vector. For the second model we make use of the symmetry functions to describe the environment and omit the DFTB1 charges of the neighbors.

Before calculating any properties with the predicted charges we have to make sure that we normalize the charges such that the sum of charges equals the charge of the molecule. For this we apply linear scaling, *i.e.*

$$q^{mol} = b\Sigma^{(+)} + b^{-1}\Sigma^{(-)}, \tag{6.6}$$

where q^{mol} is the total charge of the molecule and $\Sigma^{(+)}$ and $\Sigma^{(-)}$ are the sums of all positive and negative predicted charges, respectively. Solving for b we obtain

$$b = \frac{q^{mol} + \sqrt{(q^{mol})^2 - 4\Sigma^{(-)}\Sigma^{(+)}}}{2\Sigma^{(+)}}. \tag{6.7}$$

The positive predicted charges are now scaled by b and the negative predicted charges by b^{-1}. After obtaining these normalized charges we can rebuild the Fock matrix, and subsequently diagonalize this, to obtain the predicted properties of the system. This additional diagonalization step is needed to make the atomic orbital coefficients consistent with the definition for the charges (equation 2.36).

6.2 Experiments & results

In this section we will start off by getting a general feel for the data set and the task at hand. After this we decide upon the representation of the environment and the exact parameter settings. Finally, we evaluate the resulting model independently for the

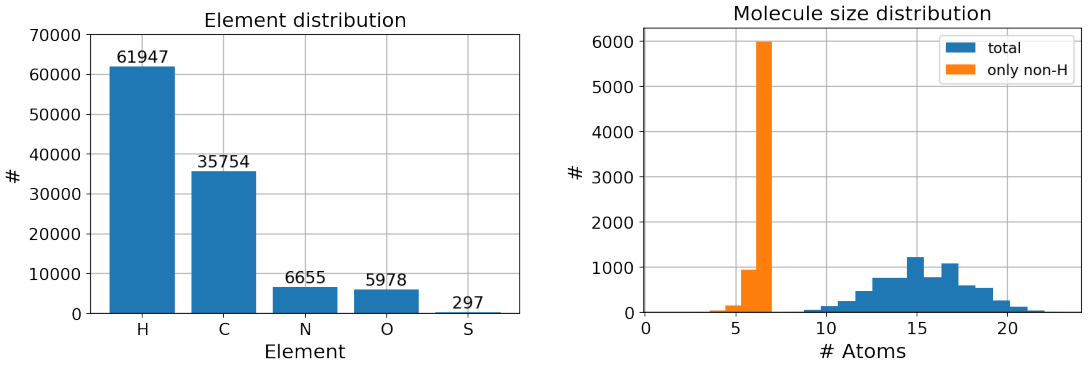

FIGURE 6.1: Distribution of element and molecule size occurrence in the **GDBP-12** data set.

different elements present in the data set, as well as the models ability to provide accurate ground state energies and generate a PES. For the experiments conducted in this section we use **ADF** [47] for the electronic structure calculations and **Keras** [44] for the NN implementation. With regards to the DFTB parameter set we use **DFTB.org/3ob-3-1** [48]. For the symmetry functions we will make use of the implementation provided in the **DScribe** Python package [49]. Unless stated otherwise we stick to atomic units for the presentation of our results [16].

6.2.1 Data set

The data set that we employ for the following set of experiments is the **GDBP-12** data set [50, 51]. This data set contains 7165 small organic molecules comprised of only the elements H, C, N, O, and S, with up to seven non-H atoms per molecule. The precise distribution of elements and molecule sizes are depicted in figure 6.1. We can see here that the data set contains predominantly H and C atoms which was to be expected since we are dealing with organic molecules. Only a very small fraction of the molecules contains S. We also find that most of the molecules contain seven non-H atoms.

Now that we know a little bit more about the build-up of the data set we are of course interested in the difference and correlation between Δq^{DFTB1} and Δq^{SCC} for the different elements. To investigate this we constructed figure 6.2. Here we can see that the C charges are usually too high in DFTB1, the O charges too low, the S charges too high, and the H charges also too high. It is now up to our models to correct this. For this we randomly select 80% of the molecular structures to form the training set and randomly divide the remaining 20% evenly over the validation and test set. We again normalize our inputs to lie within the interval [0,1], based on their range in the training set.

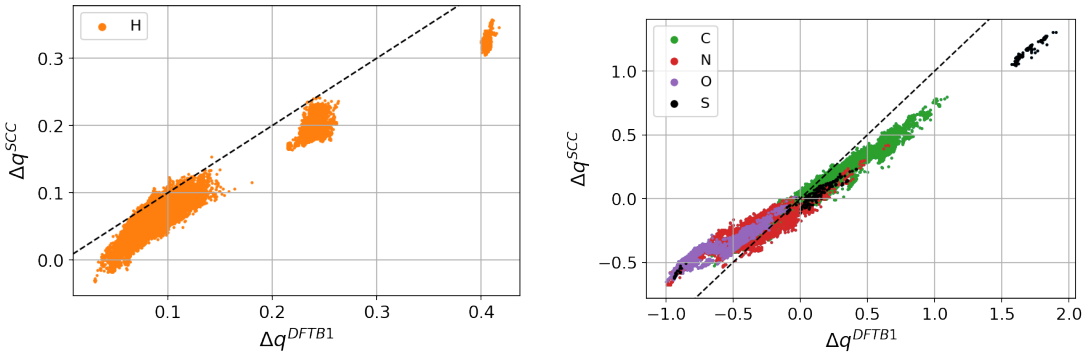

FIGURE 6.2: Correlation between Δq^{DFTB1} and Δq^{SCC} for each element in the **GDBP-12** data set. The dashed line represents $y = x$.

6.2.2 Finding the optimal representation of the atomic environment

In section 6.1.1 we discussed the two ways of defining the environment of a single atom. We now want to evaluate the performance of a NN supplied with atomic environments from either of the two methods, using different parameter settings. For the simple case of providing the relative coordinates in real space (**nosymm**) we can simply vary the amount of neighbors we take into account, again determined by parameter K (section 5.2.1). For **nosymm2** we do the same, however we enhance only the training set with four additional random rotations for each molecule. For **nosymm1** we stick to only a single random orientation per molecule. For the symmetry functions it is not as simple. In this case we decided to test three distinct parameter sets:

- **symm1**: As recommended by Zhu *et al.* [11] the first parameter combination that we consider constitutes of taking R_s to be evenly spread between 0.5 and 5.0 Å, in 24 steps, with corresponding $\eta = \frac{1}{2R_s^2}$. For the angular symmetry functions we use all unique combinations of $\omega = [0.001, 0.01, 0.05]$, $\zeta = [1, 4, 6]$, and $\lambda = [-1, 1]$.
- **symm2**: For the second set of symmetry functions we decrease the size of **symm1** such that we reduce the number of steps for R_s from 24 to 6 and instead use $\omega = [0.001, 0.01]$, $\zeta = [1, 6]$, and $\lambda = [-1, 1]$.
- **symm3**: Lastly, we also consider the standard **DScribe** settings. For the radial symmetry functions these are $R_s = 1$ and $\eta = [1, 2, 3]$ and for the angular symmetry functions $\omega = 1$, $\zeta = [1, 2]$, and $\lambda = [-1, 1]$.

We would now like to now which of these parameter settings works the best, as well as the optimal value for R_c.

For each of the methods and parameter settings we train three separate NNs on our training set. For the hyper-parameters we use the same settings as were used for the

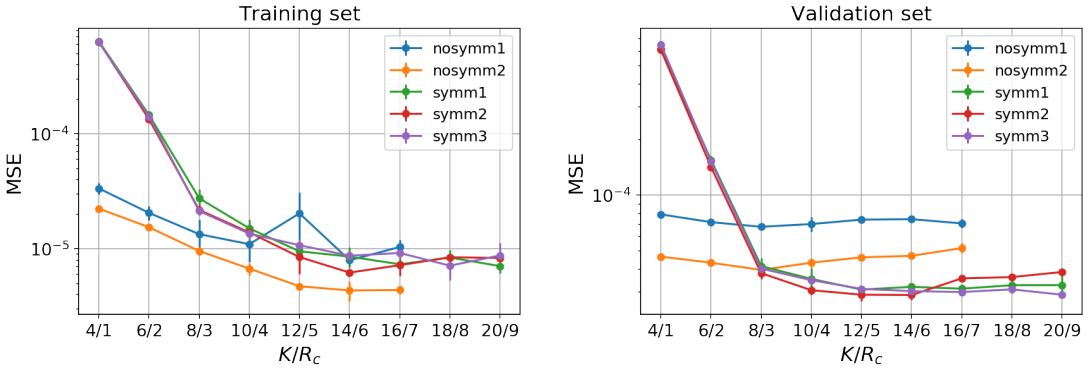

FIGURE 6.3: Training (left) and validation (right) set MSE of the trained NNs for the different atomic environment description methods varying K and R_c. *e.g.* 10/4 indicates $K = 10$ and $R_c = 4$. Depicted values are averages of three separate training sessions with error bars portraying the 95% C.I.

water model with random orientations, *i.e.* three hidden layers with 250 hidden nodes, a mini-batch size of 500 samples, and a learning rate of 0.001 applied to the Adam optimization algorithm. The model performance is evaluated using the MSE. With regards to the remaining Adam parameters we again utilize the default values proposed by [33]. As a small change we add a final hidden layer of 30 nodes to compress the data before reaching the final node [52]. We again let the model train for a maximum of 1e5 iterations and employ early-stopping, with the patience parameter set at 500 iterations. In between the hidden layers we employ the ReLU activation function and use a linear activation function at the final node. The resulting training and validation set errors are shown in figure 6.3. Looking at the training set error we see that **nosymm1** performs rather similarly to the symmetry functions. **Nosymm2** even outperforms them. However, looking at the validation set errors we see that the use of symmetry functions seems to result in a better general performance. Based on the latter, we select **symm2** with an R_c of 6 Å, as this is where we observed the lowest validation set MSE.

6.2.3 Model evaluation

The resulting model can now be evaluated based the MAE of the predicted charges with respect to SCC-DFTB. This is summarized in table 6.1 for the test set. Here we can see that the model seems to favor the more prevalent elements (H and C) with respect to absolute performance, however for the less prevalent elements (N, O, and S) a larger factor of improvement is achieved. A visual representation of the model performance is shown in figure 6.4. Here we can see that the predicted values seem to be centred around the diagonal line. This means that we have successfully removed the systematic error between DFTB1 and SCC-DFTB, and are now only left with random errors.

TABLE 6.1: MAE of the predicted charges with respect to SCC-DFTB evaluated on the test set.

	DFTB1	**NN**	**Improved by factor**
All elements	5.62e-2	3.41e-3	16.5
H	2.79e-2	2.41e-3	11.6
C	5.64e-2	4.51e-3	12.5
N	1.59e-1	5.13e-3	31.0
O	2.33e-1	5.02e-3	46.4
S	2.25e-1	9.56e-3	23.5

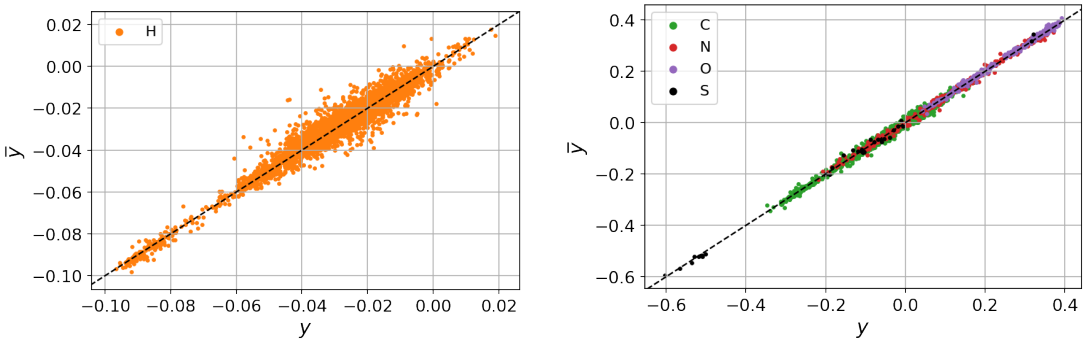

FIGURE 6.4: Required charge corrections y compared to the predicted charge corrections $\bar{y}$ for each element, evaluated on the test set. The dashed line represents $y = x$.

6.2.4 Ground state energies

Next, we are interested in what effect the resulting charge corrections have on the the properties of the system. As a measure for this we again use the ground state energy of the molecule. Before calculating the ground state energy from our predicted charges we apply linear scaling (equation 6.6). ΔE is now calculated for all the molecules in the test set and compared to the convergence of the SCC procedure (figure 6.5). Since we require an additional diagonalization step (section 6.1.2) to obtain our final result, we can directly compare our result to the second iteration (DFTB1 +1). We find that our model achieves a ΔE close to the fifth iteration, likely reducing the amount of needed SCC iterations by two or three.

6.2.5 Potential energy surface

Finally, we can take a look at a potential energy surface produced from the NN corrected charges. For this we employ ethanol as a model system. We make sure that no conformations or isomers of ethanol are present in the training or validation set. We now evaluate the PES produced by DFTB1, DFTB1 +1, the model, and SCC-DFTB. Here we consider the PES of rotating the molecule along the C-C bond. The results are depicted in figure 6.6 for both E and ΔE. Here we can see that the NN predicted PES

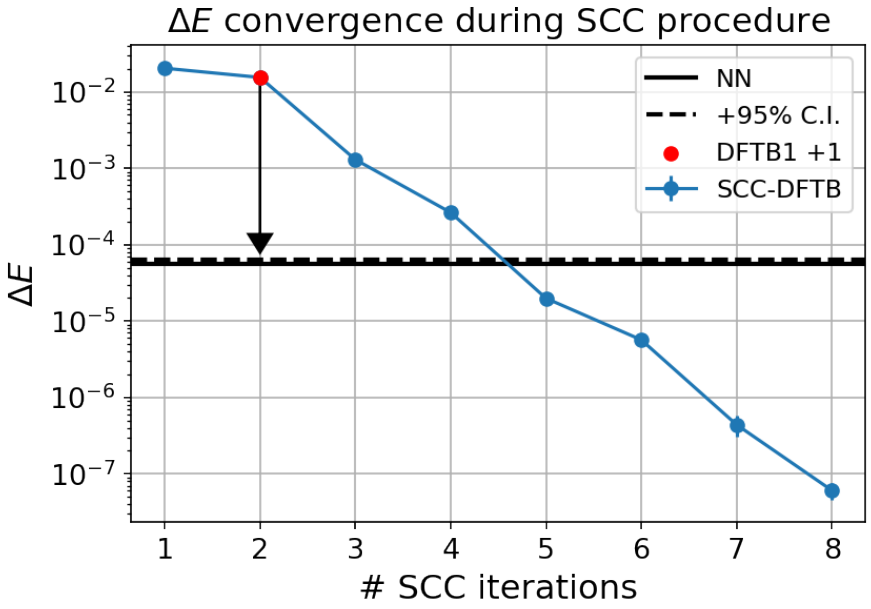

FIGURE 6.5: ΔE progression during the SCC procedure compared to the model performance. Depicted values are averages of the entire test set with error bars portraying the 95% C.I.

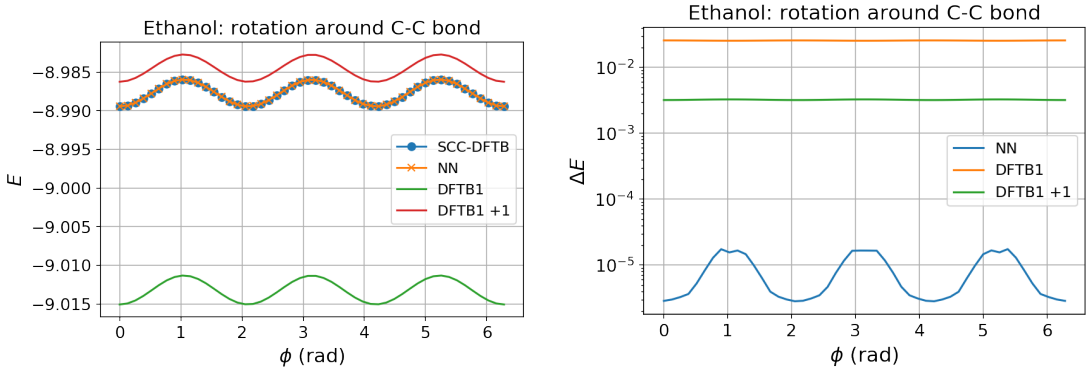

FIGURE 6.6: The PES of rotating ethanol around its C-C bond generated from the different flavors of DFTB along with our model prediction. Both E (left) and ΔE (right) are shown.

coincides with the real SCC-DFTB PES up to within chemical accuracy.

6.3 Discussion

In this chapter we took a detour from the DFT corrections and instead focused on the map between DFTB1 and SCC-DFTB, where only SCC-DFTB requires a self-consistent procedure. This simplified the corrections quite a bit, since this mapping only a requires a set of rotationally invariant net Mulliken charges to obtain the SCC-DFTB Fock matrix from its DFTB1 counterpart. Instead of targeting a single molecule we attempted to generalize our predictions to small organic molecules containing only H, C, N, O, and S. This deviated from the earlier approach [14] in that we now started off by doing a DFTB1 calculation and used the resulting charges in our model, both as an input and a basis for the corrections, in contrast to predicting these charges based solely on the molecular geometry. A downside to this is that it requires an additional diagonalization step to make the MOs consistent with the charges.

In order to correct the charges we made use of a single NN, resorting to roughly the same set of hyper-parameters as used for the water molecule with random orientations. In order to explore the full potential of such a model this would of course require further hyper-parameter tuning. The main challenge here constituted of finding the optimal way of representing the environment. We found that by employing symmetry functions to describe the environment we improved the ability of our model to make general predictions, without the need of supplying multiple orientations of the same molecule. This, compared to the more simple way of representing the environment which does require these added orientations to improve upon prediction power. For the latter, one could try to add even more randomly rotated geometries to see if the results improve. However, the symmetry functions still have the advantage of yielding the same prediction, independent of the supplied rotation of your structure. A good point of improvement might be to add back the DFTB1 charges of the surrounding atoms, for the symmetry function based input, as more data for the model to work with.

After deciding on the set of symmetry functions we evaluated the model based on its performance for each of the elements. With regards to the absolute performance the model seemed to do better for the more prevalent elements H and C. However, looking at the factor of improvement we observed big improvements for N, and especially O. With regards to S we also observed a big improvement factor, but still an absolute error of about two to three times that of the other elements. This was likely caused by the small fraction of S contained in the data set. To improve upon this, one could train dedicated NNs specialized at correcting the charge of a single element [14]. Another way to improve the performance is of course to use a larger data set. However, comparing the predicted charges to the real charges we seemed to have successfully removed the systematic DFTB1 to SCC-DFTB error. Next, we looked at the convergence of the ground

state energy during the SCC procedure and compared this to our model performance. As stated earlier, our approach required an additional diagonalization step, so we can effectively compare our result to the second SCC step. Based on these results, we hypothesize that one can effectively skip two to three SCC steps by applying our approach to similar use cases. However, for such small molecules this will likely not amount to any noticeable speedup. This, because the part of the Fock matrix containing the complicated integrals only needs to be calculated once. Only when the diagonalization step becomes the rate-determining step, we can expect a significant speedup. With regards to generating a PES we found that the ground state energy was well reproduced along the rotation of ethanol around its C-C bond.

Chapter 7

Generalized Fock matrix corrections

Now that we have all tools in place we can finally take on the challenge of attempting to generalize DFT Fock matrix level corrections to small organic molecules. For this purpose, we start off by dealing with the size dependence of the Fock matrix on the size of the molecule, after which we define our model. The model will then be evaluated in a similar manner as done in the previous two chapters.

7.1 Method

In section 5.2.1 we realized that we can split the Fock matrix into different parts describing different atomic interactions. For a single molecule, such as water, this is of course not necessary. However, in order to be able to generalize between different molecules this is essential. This, because the size of the Fock matrix, and thus the amount of input/output variables for a model to process, is dependent on the size of the system. The model described in section 5.2.1 can in theory be applied to this problem, however we decided to take a more specialized approach here.

7.1.1 Model description

First off, instead of correcting the Fock matrix we change our perspective to correcting only the density dependent part $\mathbf{B}[\rho]$ such that

$$\mathrm{F}_{ij} = \langle \chi_i | \hat{h}^{core} | \chi_j \rangle + \mathrm{B}_{ij}[\rho], \tag{7.1}$$

since this is the only part that changes during the SCF procedure. The correction matrix $\mathbf{Y}$ remains unchanged, since

$$\mathrm{Y}_{ij} = \mathrm{F}_{ij}^{DFT} - \mathrm{F}_{ij}^{guess} = \mathrm{B}_{ij}^{DFT} - \mathrm{B}_{ij}^{guess}, \tag{7.2}$$

where $\mathbf{B}^{DFT}$ and $\mathbf{B}^{guess}$ are the density dependent parts of $\mathbf{F}^{DFT}$ and $\mathbf{F}^{guess}$, respectively. $\mathbf{F}^{guess}$ and $\mathbf{B}^{guess}$ are again constructed from the superposition of free-atom densities. The only practical implication of this change is that we use the elements from $\mathbf{B}^{guess}$ as model inputs instead of the ones from $\mathbf{F}^{guess}$, effectively removing the invariant single-electron contributions. Once again we stick to the minimal basis set STO-6G and will consider only the elements: H, C, N, and O. In STO-6G, H is described by a single $1s$-type basis function and C, N, and O by five basis functions of type $[1s, 2s, 2p_x, 2p_y, 2p_y]$. The next change is that we define multiple models to do the corrections. The following list describes all fourteen separate models **NN#** we intend to use along with the information we provide and the corrections we aim to make:

- **NN1**: Here we correct element H-$1s$. For this we provide the atomic environment and the $\mathbf{B}^{guess}$ element value.
- **NN2**, **NN3**, **NN4**: Here we correct the diagonal elements of the C/N/O self-interaction block divided over the three NNs (one for each element). For this we provide the atomic environment and the value of the corresponding elements in $\mathbf{B}^{guess}$.
- **NN5**, **NN6**, **NN7**: Here we correct the off-diagonal elements of the C/N/O self-interaction block divided over the three NNs (one for each element). For this we provide the atomic environment and the value of the unique upper triangular elements of the C/N/O self-interaction block in $\mathbf{B}^{guess}$.
- **NN8**: Here we correct the H-$1s$ interaction with another H-$1s$ basis function. For this we provide the atomic environment of both Hs, the distance r between both atoms, and the value of the corresponding element in $\mathbf{B}^{guess}$.
- **NN9**: Here we correct the H-$1s$ interactions with all five basis functions of C, N, and O. For this we provide the atomic environment of H, the interacting atom-type out of [C, N, O] as a one-hot encoded vector, the environment of the interacting atom, the distance r between both atoms, the normed distance vector $\hat{\mathbf{r}}$, and the value of the corresponding elements in $\mathbf{B}^{guess}$.
- **NN10**, **NN11**, **NN12**, **NN13**, **NN14**: Here we correct the interactions of the $1s/2s/2p_x/2p_y/2p_z$ basis functions centred on [C, N, O] with all the basis functions of a neighboring [C, N, O] divided over five NNs (one for each basis function type).

For this we provide the atomic environment of both atoms, both of their types as a one-hot encoded vector, the environment of both atoms, the distance r between both atoms, the normed distance vector $\hat{\mathbf{r}}$, and the value of the corresponding elements in $\mathbf{B}^{guess}$.

For each of the models we use symmetry functions (section 6.1.1) to describe the environment. The relative orientation of interacting atoms is now solely described by the combination of r and $\hat{\mathbf{r}}$, and rotationally variant elements of $\mathbf{B}^{guess}$. Note that we do not provide $\hat{\mathbf{r}}$ for H-H interactions since these are invariant under rotations. From the predictions of all NNs we can finally construct $\bar{\mathbf{Y}}$, after which we symmetrize (equation 5.2) before obtaining the final corrected Fock matrix (equation 5.3). For the corrections we limit ourselves to the K nearest neighbors of each atom.

7.2 Experiments & results

In this section we will revisit the data set used in chapter 6, enhance it, and perform an additional analysis based on the current challenge. After this we decide upon the amount of interactions we aim to correct, determined by parameter K, and finally do an extensive evaluation of the model performance. For the experiments conducted in this section we use **PySCF** [43] for the electronic structure calculations and **Keras** [44] for the NN implementation. For the symmetry functions we again make use of the implementation provided in the **DScribe** Python package [49]. Unless stated otherwise we stick to atomic units for the presentation of our results [16].

7.2.1 Data set

For the following set of experiments we again resort to the **GDBP-12** data set (section 6.2.1). However, before using this data set we disregard all molecules containing S. Next, we enhance this data set with random perturbations of the geometry for each molecule. This entails that we add a normally distributed offset to each of the Cartesian atomic coordinates with mean 0 and standard deviation 0.1 Å. When this results in a non-converging DFT calculation we disregard only the perturbed molecule.

We can now take a look at the data set descriptors with respect to the differences between $\mathbf{F}^{DFT}$ and $\mathbf{F}^{guess}$. For this we use the absolute difference between the two for single elements $|\Delta \mathrm{F}_{ij}^{guess}|$ and the MAE evaluated on the entire matrix MAE $\mathbf{F}^{guess}$ as measures for this difference. The relation between the individual elements is summarized in figure 7.1. From this we can tell that elements of the two matrices are highly correlated. We

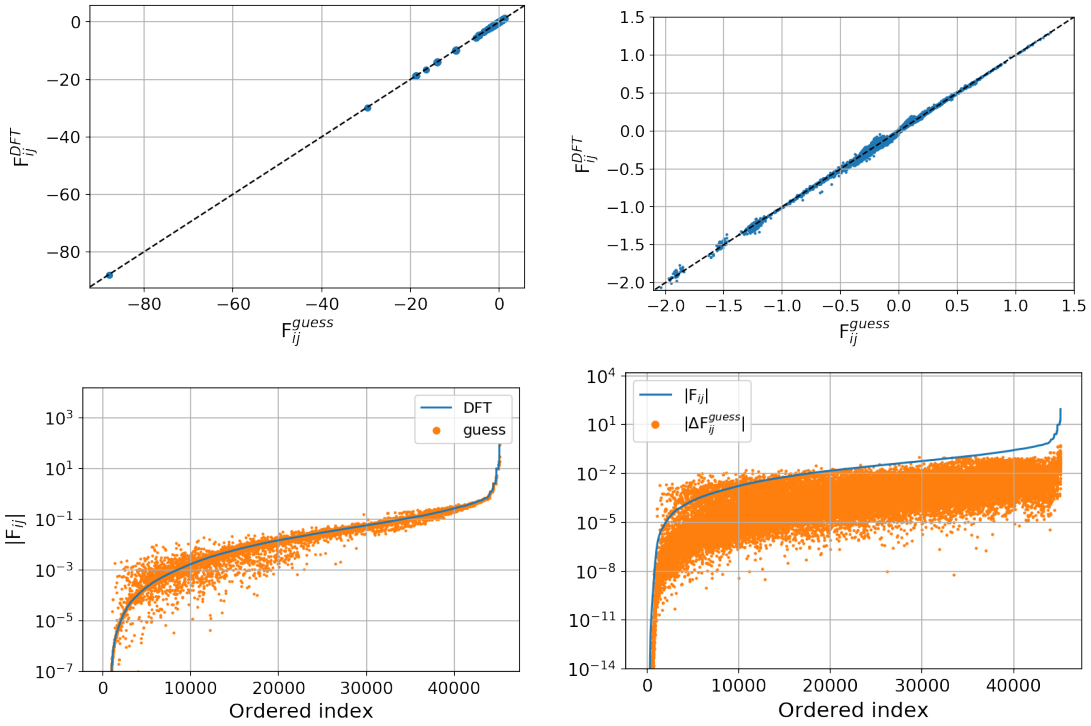

FIGURE 7.1: Correlation between F_{ij}^{DFT} and F_{ij}^{guess}, their absolute values, and their absolute difference $|\Delta F_{ij}^{guess}|$. The dashed line represents $y = x$. Ordered index indicates that the points represented by the blue line are ordered from low to high, with corresponding points represented as scatter points at the same point on the horizontal axis. Depicted plots represent 50 randomly selected molecular structures from the data set.

TABLE 7.1: Description of the data set with regards to the defined molecular subgroups.

Molecular group	**# Structures**	**Prevalence (%)**
Total	14313	100
C,H	1022	7.14
+N	4124	28.8
+O	3872	27.1
+N,O	5295	37.0

also observe an increase in absolute difference for absolute larger matrix elements.

To get a feel for the effect that the molecular buildup has on the difference between the matrices we divide the data set into four groups. These groups are: molecules containing only H and C **[H,C]**, molecules containing N **[+N]**, molecules containing O **[+O]**, and molecules containing both N and O **[+N,O]**. From figure 7.2 we can see that the presence of elements N and O results in larger required corrections and a more widespread distribution in MAE $\mathbf{F}^{guess}$. The exact division of the data set in molecular subgroups is specified in table 7.1. Finally, we divide the data set into a training set, formed from 80% of the molecules, and a validation and test set, each including 10% of the molecules. We make sure that the original molecule and its corresponding randomly

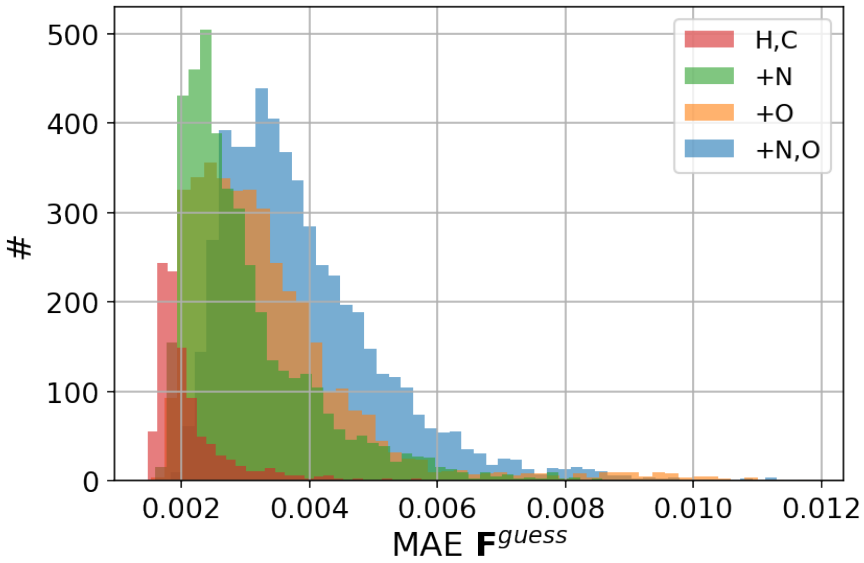

FIGURE 7.2: Distribution in MAE $\mathbf{F}^{guess}$ for the different molecular subgroups.

perturbed structure belong to the same subset, such as to not create any unwanted biases. Once again we normalize our inputs to lie within the interval [0,1], based on their range in the training set.

7.2.2 Model parameters

For the model parameters we again stick to three hidden layers, a mini-batch size of 500 samples, and a learning rate of 0.001 applied to the Adam optimization algorithm, with the remaining Adam parameters once again set to values values proposed by [33]. However, to minimize overfitting and improve the convergence we decrease the size of the hidden layers from 250 to 70, 60, and 50 for the first, second, and third hidden layer, respectively. Just as before we let each of the models train for a maximum of 1e5 iterations and use early-stopping to minimize overfitting with the patience parameter set to 500 iterations. In between the hidden layers we employ the ReLU activation function and for the final nodes we use a linear activation function. The model performance is again evaluated using the MSE. Based on the experiment presented in section 6.2.2 we stick to **symm2** with an R_c of 6 Å.

What remains now is to determine the parameter K that dictates how many of the nearest neighbour interactions we train on and will attempt to correct. For this we replaced the entries of $\mathbf{F}^{guess}$ corresponding to K nearest neighbour interactions with their $\mathbf{F}^{DFT}$ counterparts. Note that for $K = 0$ we only replace the intra-atomic interaction terms. The results are depicted in figure 7.3 for ΔE and the MAE of the resulting MO energies/eigenvalues MAE $\boldsymbol{\varepsilon}$. Shooting for chemical accuracy, and a little extra, we

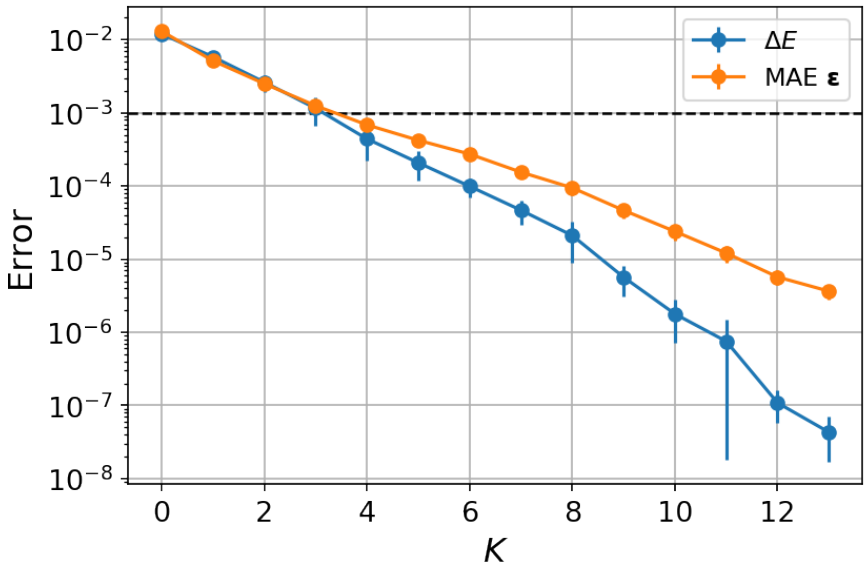

FIGURE 7.3: Convergence of ΔE and MAE ε when increasing K. Dashed line indicates chemical accuracy. Data points represent averages from 100 random structures in the data set with at least 14 atoms. Error bars represent 95% C.I.

decide upon $K = 7$.

7.2.3 Model evaluation

After the training procedure we can evaluate the performance of our model in a number of ways by employing the test set. First, we look at the individual performance of each separate NN with respect to the correction of individual elements F_{ij} displayed in table 7.2. Here we can see that the combined NNs reduce the absolute error for F_{ij} on average by a factor of ~ 3.

Next, we revisit the molecular groups that we defined before. Looking at table 7.3 we find that for each of the groups we significantly improve upon the initial guess, again by a factor of ~ 3. The performance of the model however still reflects the initial error, *e.g.* the error for group **[+N,O]** is still the largest and for **[H,C]** still the lowest. This, even though the **[+N,O]** group makes up the largest fraction of the data set.

In order to learn more about the performance of the model for the individual matrix elements we constructed figure 7.4 in which we compare our absolute prediction error $|\Delta \mathrm{F}_{ij}^{NN}|$ to $|\Delta \mathrm{F}_{ij}^{guess}|$. Here we observe that our model not only improves upon the guess, but also worsens it in many cases. However, looking at both distributions we still observe a global shift to a more accurate result.

Finally, we take a look at a similar plot comparing the MAE of the predicted Fock matrix MAE $\mathbf{F}^{NN}$ to MAE $\mathbf{F}^{guess}$ (figure 7.5) for each of the molecular subgroups. We

TABLE 7.2: MAE of the predicted Fock matrix entries by each individual NN with respect to DFT, evaluated on the test set.

NN#	Guess	NN	Improved by factor
Total	4.61e-3	1.54e-3	2.99
1	1.41e-2	6.95e-3	2.03
2	3.99e-2	7.76e-3	5.14
3	3.63e-2	1.17e-2	3.10
4	3.97e-2	1.42e-2	2.80
5	4.27e-3	1.80e-3	2.37
6	1.31e-2	3.98e-3	3.29
7	1.70e-2	3.86e-3	4.40
8	1.04e-3	5.63e-4	1.85
9	1.76e-3	9.25e-4	1.90
10	8.32e-4	1.60e-4	5.20
11	3.22e-3	1.32e-3	2.44
12	2.39e-3	1.21e-3	1.98
13	2.58e-3	1.22e-3	2.11
14	2.57e-3	1.22e-3	2.11

TABLE 7.3: MAE of the predicted Fock matrices with respect to DFT, evaluated on the test set containing in total 1373 molecular structures.

Molecular group	Prevalence (%)	Guess	NN	Improved by factor
Total	100	3.27e-3	1.10e-3	2.97
H,C	7.57	2.17e-3	7.55e-4	2.87
+N	24.8	2.89e-3	1.01e-3	2.86
+O	29.4	3.20e-3	1.03e-3	3.11
+N,O	38.3	3.78e-3	1.29e-3	2.93

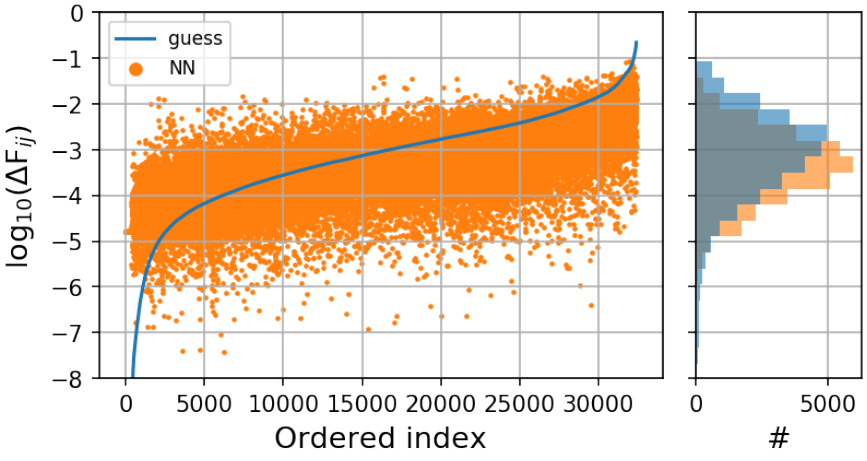

FIGURE 7.4: $|\Delta F_{ij}^{NN}|$ and corresponding $|\Delta F_{ij}^{guess}|$. Points were ordered according to increasing $|\Delta F_{ij}^{guess}|$. Depicted values correspond to 50 randomly selected molecular structures from the test set.

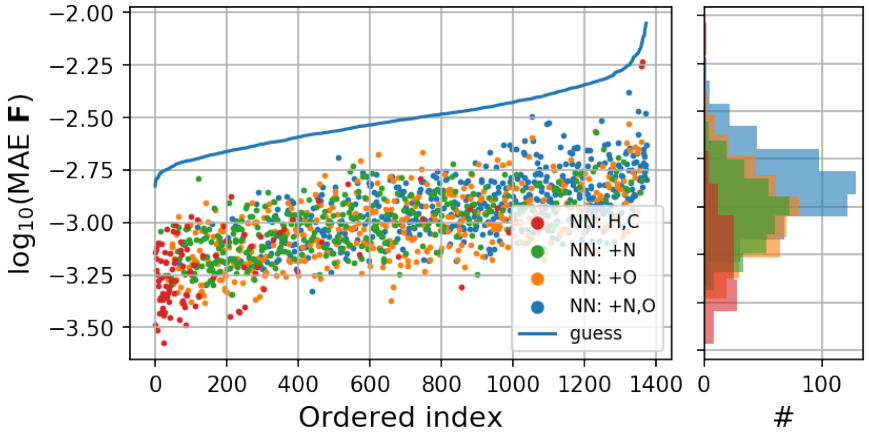

FIGURE 7.5: MAE $\mathbf{F}^{NN}$ and corresponding MAE $\mathbf{F}^{guess}$ evaluated on the test set. Points were ordered according to increasing MAE $\mathbf{F}^{guess}$ and divided into their molecular subgroups. Blue distribution corresponds to [+**N,O**].

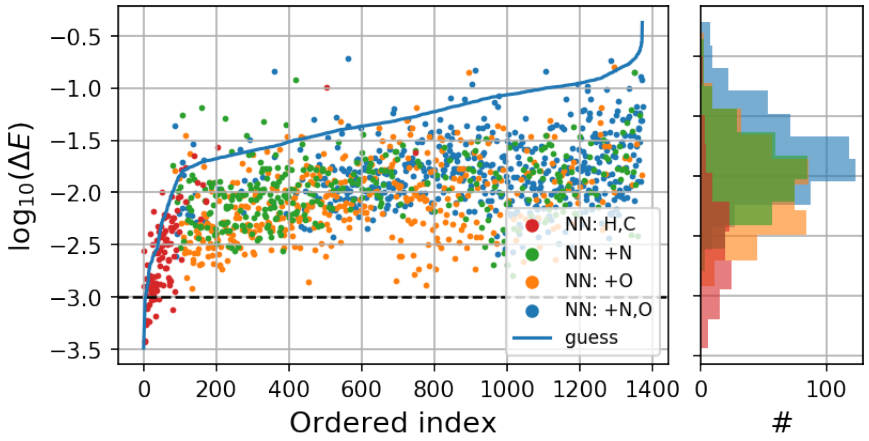

FIGURE 7.6: ΔE^{NN} and corresponding ΔE^{guess} evaluated on the test set. Points were ordered according to increasing ΔE^{guess} and divided into their molecular subgroups. Blue distribution corresponds to [+**N,O**]. Dashed line indicates chemical accuracy.

can clearly see here that none of corrected Fock matrices have become worse, according to this metric. We also observe that a bad initial guess results in a larger remaining error after corrections.

7.2.4 Ground state energy & molecular orbital energies

Just as in the previous two chapters we also consider the predicted ground state energies for each of the molecules in the test set and judge them based on ΔE. We can look at the results for each molecular subgroup in figure 7.6. We again observe a similar trend in performance between the subgroups. In almost none of the cases chemical accuracy is achieved. With regards to the MAE ε (figure 7.7) the spread seems to be slightly more uniform across the groups.

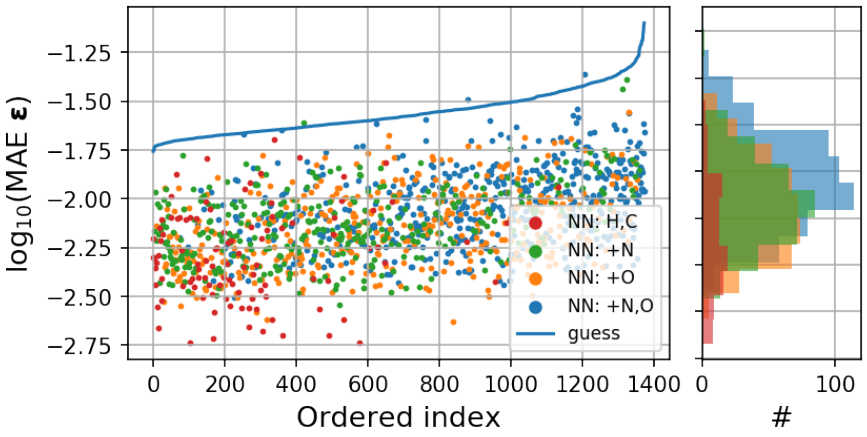

FIGURE 7.7: MAE ε^{NN} and corresponding MAE ε^{guess} evaluated on the test set. Points were ordered according to increasing MAE ε^{guess} and divided into their molecular subgroups. Blue distribution corresponds to **[+N,O]**.

7.2.5 Potential energy surface

Based on the results in the previous section one might already conclude that this model is not fit for constructing a PES, due to the errors present in the predictions for the ground state energies. However, constructing a PES is still an interesting way to see how well the model deals with rotations of the molecule. For this purpose we revisit our model system ethanol. Again we make sure no conformations or isomers of ethanol are present in the training or validation set. Just as before, we rotate ethanol around its C-C bond to form a PES. Since the Fock matrix is not invariant under rotations, we generate one PES where we align the C-C bond with the z-axis and rotate only one of the C atoms along with its bonds. We also evaluate the PES in the case where we give the molecule a random orientation in space after rotating along the C-C bond. The results are shown in figure 7.8. Here we find that with regards to the absolute ground state energy we improve upon the guess along the entire PES. Considering now only the corrected PES for a stationary C-C bond, we find that the shape of the PES still seems to be preserved after doing the corrections. However, for the randomly oriëntated PES this is not the case, as it looks very noisy.

7.2.6 Accelerating SCF convergence

Let us now see if we can apply our model in a different way, namely to reduce the amount of SCF iterations needed for convergence. For this we consider the three available methods for generating the initial 1RDM in **PySCF** [43], namely **minao**, **hcore**, and **atom**. **Minao** relies on the use of an atomic natural orbital basis to obtain a starting electron-density. This density is then projected onto the employed basis set to generate

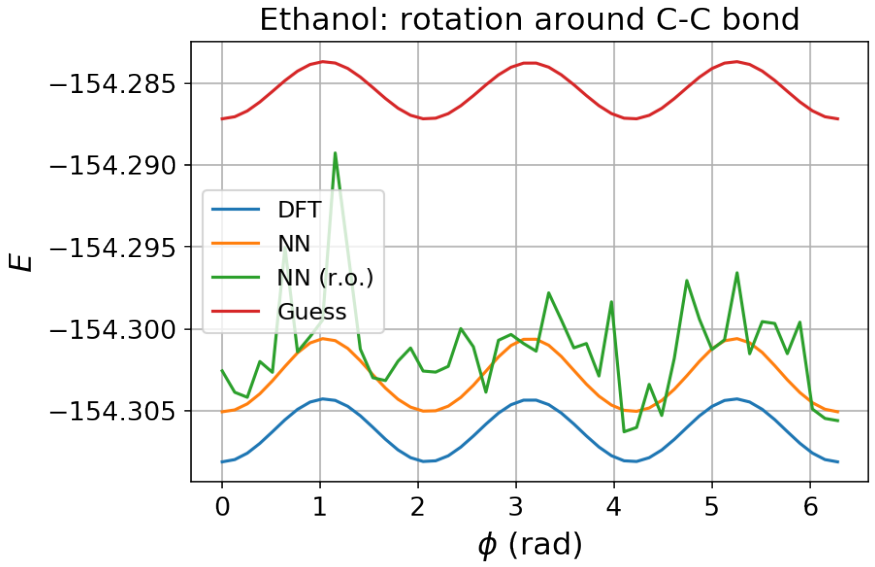

FIGURE 7.8: PES of rotating ethanol around its C-C bond. (r.o.) Indicates that we give the molecule a random orientation before evaluating E. The other NN generated PES is formed by aligning the C-C bond with the z-axis and rotating one of the sides.

the first guess for the 1RDM. **Hcore** generates this first guess by minimizing only the density-independent part of the energy, requiring only a single diagonalization. We did not include this in our results, because generally the amount of iterations required and time needed for convergence far exceeded the other three. In addition to this, we also experienced convergence issues using this method. **Atom** corresponds to the initial guess based on the superposition of free-atom densities, treated in section 4.2. This is what is used as the basis for the NN corrections. In **PySCF** the SCF convergence is evaluated based on the absolute difference in ground state energy between succeeding iterations and the norm of the orbital gradients [53]. With regards to this, the thresholds are set to 1e-10 and 1e-5, respectively.

After obtaining the initial 1RDM, for each of these methods, we generate the first guess for the Fock matrix. We label this step, plus optional NN corrections, as the initialization procedure. The initial Fock matrix is then diagonalized, and the resulting 1RDM is fed back into **PySCF** to start the SCF procedure. The resulting calculation times and number of SCF calculations are reported in figure 7.9. Looking at this figure, we find that using the NN corrected Fock matrix we obtain a slight reduction in SCF time from 6.52 to 5.84 s and a small increase in initialization time, accounting for the NN evaluation, from 2.09 to 2.31 s. Taking the sum of the two we end up with a net speed-up of 0.47 s or 5.46%. More interesting however is the reduction of required SCF iterations from an average of 11.63 to 10.24 cycles. This corresponds to an improvement of 1.39 cycles or 12.0% with respect to **atom**.

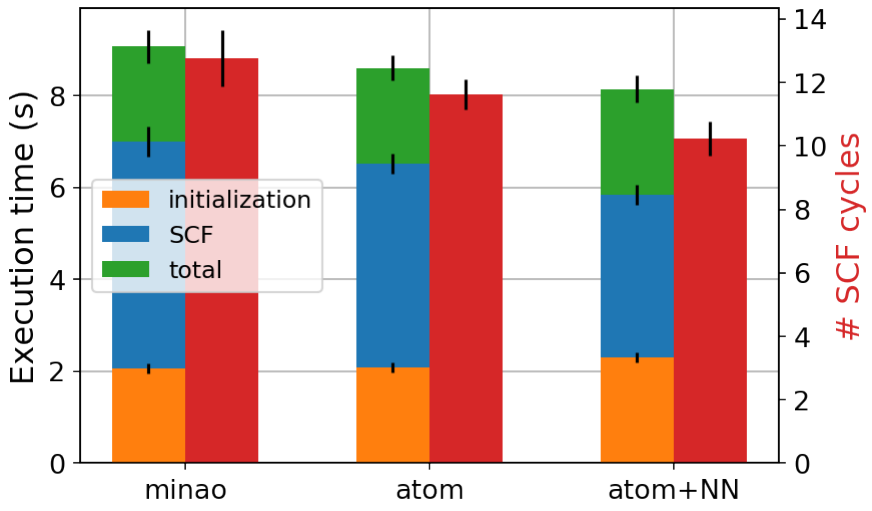

FIGURE 7.9: Time spent on the DFT calculation as well as the number of SCF iterations required for convergence for different initialization methods. Reported values are averages of 100 random structures present in our test set. Error bars depict 95% C.I.

7.3 Discussion

After we learned something about Fock matrix corrections for a single molecule and how different orientations of the molecule affected this, as well as how to efficiently describe the atomic environment, we took on the challenge described in this chapter. Here, we attempted to predict the converged DFT Fock matrix of small organic molecules (comprised of only the elements C, H, N, and O), again by using the initial guess based on free-atom densities as a basis. However, this time we could not simplify the corrections to something like a set of charges, as we did for DFTB, but instead resorted to correcting each element individually. To achieve this, we employed an enhanced version of the data set used in chapter 6, however we removed any molecules containing S to simplify the task. We then constructed in total fourteen NNs, each of them dedicated to a subset of Fock matrix elements. We slightly decreased the size of these networks as compared to the one used for water with random orientations. This, because we observed stability and overfitting issues for this architecture, even after a few iterations. To explore the full potential of this model one of course need to tune the hyper-parameters individually for each of the NNs. We have yet to attempt this since a single training run quickly amounted to upwards of 12 hours. We also decided to go with the same set of symmetry functions as for DFTB which could also be tuned specifically for this purpose.

After we defined our model and carried out the necessary calculations, we went on to analyze the resulting data set. We found that the value of a given element in the guess for Fock matrix is largely correlated to its converged counterpart. We were therefore dealing with very small corrections, as compared to the previously treated charges. Next, we divided the data set into four subsets and concluded that the most prevalent subgroup, molecules containing both N and O, required the largest amount of correcting, as opposed to molecules containing neither, which required the least. This, likely because of the shift in electron-density, with respect to the free-atom densities, due to the electronegativity of these elements [54]. After this we determined that we only consider a set number of nearest neighbor interactions for our corrections, namely seven neighboring atoms, as this would be enough to achieve chemical accuracy with regards to the ground state energy and the MO energies. To be more consistent with the cut-off radius for the symmetry functions, one could also opt to define a radius within which to consider atomic interactions.

After settling on all the parameters and training the different NNs we went on to analyze our results. On average we observed an improvement by a factor of ~ 3 with respect to the guess matrix elements. From the NN specific errors we learned that even with the use of symmetry functions the rotationally invariant elements, such as the H-1s diagonal element and the H-H interaction elements, still posed quite a challenge. This

likely means that we need to increase the amount of molecular structures in the data set. However, this could also mean that symmetry functions are simply not suited for predicting such complex matrix elements. Looking at the guess and correction errors for individual elements we observed that the model also worsens the guess in many cases. Finding a way to identify these pathological cases might help to improve the model, possibly through clever usage of another type of initial guess for the electron-density. Next, we evaluated our model performance more globally based on the MAE of the entire predicted Fock matrix. Here we observed a similar factor of improvement for each of the defined molecular subgroups of ~ 3. We also found that none of the Fock matrices had become worse, based on this metric. With regards to the ground state energy we again observed a similar distribution in prediction error amongst the subgroups, with chemical accuracy only being achieved for a small fraction of the molecules. With regards to the eigenvalues the observed prediction errors were a little more uniformly divided amongst the groups.

Next, we applied our model to two different use cases. We started off by revisiting our experiment of generating a PES for the rotation of ethanol around its C-C bond. We did this for both a stable orientation and random orientations. We found that for the case of a stable orientation the shape of the PES seemed well preserved, in contrast to the randomly rotated case which contained a lot of noise. This tells that the model still struggles a lot with these rotations. This means that, for future modelling, the data set likely needs to be enhanced with multiple rotations of each training sample. Again we propose the use of Wigner rotation matrices to avoid redoing any calculations. From these results we can conclude that, in this state, the model is unfit for producing any chemically meaningful results on its own, especially if we consider that we require significantly larger basis sets to obtain chemically useful DFT results in the first place [55]. However, even though our corrected Fock matrices are not quite accurate enough to serve as a final result, we found that by using them as a starting point for the SCF procedure, we could reduce the amount of SCF iterations required for convergence by 1.39 cycles or $\sim 12\%$. Extrapolating this result to larger basis sets and molecules this would correspond to a substantial reduction in computation time. The main advantage of this method is that it can in principle be applied to any initial guess for the Fock matrix, such as the one presented in [41]. A downside however, is that it requires re-training for the inclusion of new elements and the extension to bigger basis sets, as well as the use of different functionals. Again, a good starting point for future research might be to try and combine our approach, of exploiting the predictive power of an initial guess for the Fock matrix, with the one presented in [13].

Chapter 8

Conclusion

In this thesis we introduced a set of different ways to apply NNs to quantum chemistry, where we mainly base the predictions on cheaply obtained and less accurate results. Let us start by stressing that the presented models and approaches have not yet been explored up to their full potential and serve more as a starting point than a final result.

In chapter 4 we started off by defining the initial guess for the Fock matrix, constructed from an electron-density comprising of free-atom densities, as a starting point for machine learning corrections. This, because it is consistent between different molecules, as the free-atom density remains unchanged. It is then up a model to redistribute this electron-density across the molecule, through correcting the Fock matrix, with the goal of replicating the converged DFT result. This approach differs from most other approaches, which attempt to directly predict a subset of molecular properties, in the fact that we retain access to all the desired properties of the system at DFT level. One of the main challenges of this approach is dealing with rotations of molecular systems and constructing a suitable description of the environment. Since this in itself already offers quite a challenge, we stuck to a minimal basis set, namely STO-6G, for all of our DFT based models. Consequently the obtained results are not suitable for real-life applications but serve more as a proof of concept.

In chapter 5 we applied this methodology to a water molecule, with the goal of exploring the performance of a NN on the PES, both within and outside the training region. Here we found that rotating the molecule posed a significant challenge to the NN. However, both in the case of a mostly stationary water molecule and one which was randomly rotated before doing the corrections, we obtained results within chemical accuracy for geometries within the training region. From this we conclude that a NN can successfully interpolate between the training data. Even up to a limited range outside the training data the model could make some sensible predictions. From this we conclude that such

an approach can be useful when studying small molecular systems in detail. For future research we propose to enhance the model presented in [13] by adding the initial guess for the Fock matrix to the set of input features and defining the predictions as a correction.

In chapter 6 we moved to modelling the mapping between DFTB1 and SCC-DFTB, which only requires correcting the net Mulliken charges on individual atoms, with the added challenge of generalizing our approach to small organic molecules containing only the elements H, C, N, O, and S. For this purpose we build upon the research presented by [14]. We found that using a NN we can correct these charges such that we obtain a chemically accurate result after a single diagonalization. To achieve this, we employed the help of rotationally invariant symmetry functions to describe the environment of all the atoms in a molecule. It turns out that, due to the rotationally invariant nature of these functions, the trained model can also be applied to generate a PES. Based on our results, we conclude that a NN is well capable of mapping DFTB1 to the more accurate SCC-DFTB. For future research we propose further hyper-parameter optimization and researching the need of supplying the DFTB1 charges to the model, both as an input and a basis for the corrections. Removing the dependence on these charges would eliminate the need for the additional diagonalization step. One could possibly achieve this through employing a larger data set.

Using the knowledge gathered in chapters 4, 5, and 6, we also attempted to generalize our DFT Fock matrix predictions to small organic molecules composed of the elements H, C, N, and O (chapter 7). For this we employed in total fourteen separate NNs. Based on numerical simulations, we conclude that the resulting Fock matrices are not fit for usage as a final result, but could serve as an improved starting point for the SCF procedure. This, based on the fact that we achieved a reduction of 1.39 cycles or $\sim 12\%$ for the structures in our test set. For future research we again propose the combination of our approach with the one presented in [13], as well as the use of Wigner rotation matrices to cheaply enhance the data set. For the current model we also recommend carrying out a hyper-parameter optimization procedure for the individual NNs. One could of course also explore different ways of dividing the Fock matrix elements between different NNs. In addition to this, it would be interesting to see how well this type of model would perform for larger basis sets, however this would likely require a much larger data set than the one used here.

Bibliography

[1] Theories of second-language acquisition. https://www.wikiwand.com/en/Theories_of_second-language_acquisition. Accessed: 7 July 2020.

[2] Ramazan Gençay and Min Qi. Pricing and hedging derivative securities with neural networks: Bayesian regularization, early stopping, and bagging. *Neural Networks, IEEE Transactions on*, 12:726 – 734, 08 2001. doi: 10.1109/72.935086.

[3] Hydrogen atom model motion - atomic orbital electron hydrogen atom quantum number, 2019. URL https://favpng.com/png_view/hydrogen-atom-model-motion-atomic-orbital-electron-hydrogen-atom-quantum-number-png/BPJ4THic. Accessed: 11 August 2020.

[4] F. Jensen. *Introduction to Computational Chemistry*. Wiley, 3 edition, 2017.

[5] Brian B. Laird, Richard B. Ross, and Tom Ziegler. *Density-Functional Methods in Chemistry: An Overview*, volume 629 of *ACS Symposium Series*, pages 1–17. American Chemical Society, May 1996. ISBN 9780841234031. doi: 10.1021/bk-1996-0629.ch001. URL https://doi.org/10.1021/bk-1996-0629.ch001. 0.

[6] Justus Thies, Michael Zollhöfer, and Matthias Nießner. Deferred neural rendering: Image synthesis using neural textures. *ACM Transactions on Graphics 2019 (TOG)*, 2019.

[7] Ye Jia, Yu Zhang, Ron J. Weiss, Quan Wang, Jonathan Shen, Fei Ren, Zhifeng Chen, Patrick Nguyen, Ruoming Pang, Ignacio Lopez-Moreno, and Yonghui Wu. Transfer learning from speaker verification to multispeaker text-to-speech synthesis. *CoRR*, abs/1806.04558, 2018. URL http://arxiv.org/abs/1806.04558.

[8] Alvaro Sanchez-Gonzalez, Jonathan Godwin, Tobias Pfaff, Rex Ying, Jure Leskovec, and Peter W. Battaglia. Learning to simulate complex physics with graph networks, 2020.

[9] Pavlo O. Dral. Quantum chemistry in the age of machine learning. *The Journal of Physical Chemistry Letters*, 11(6):2336–2347, 2020. doi: 10.1021/

acs.jpclett.9b03664. URL https://doi.org/10.1021/acs.jpclett.9b03664. PMID: 32125858.

[10] Raghunathan Ramakrishnan, Pavlo O. Dral, Matthias Rupp, and O. Anatole von Lilienfeld. Big data meets quantum chemistry approximations: The δ-machine learning approach. *Journal of Chemical Theory and Computation*, 11(5):2087–2096, May 2015. ISSN 1549-9618. doi: 10.1021/acs.jctc.5b00099. URL https://doi.org/10.1021/acs.jctc.5b00099.

[11] Junmian Zhu, Van Quan Vuong, Bobby G. Sumpter, and Stephan Irle. Artificial neural network correction for density-functional tight-binding molecular dynamics simulations. *MRS Communications*, 9(3): 867–873, 2019. URL https://www.cambridge.org/core/journals/mrs-communications/article/artificial-neural-network-correction-for-densityfunctional-tightbinding-molecular-dynamics-simulations/F4901D708C0D5D19BFA24A708678B99A.

[12] Kevin Ryczko, David A. Strubbe, and Isaac Tamblyn. Deep learning and density-functional theory. *Phys. Rev. A*, 100:022512, Aug 2019. doi: 10.1103/PhysRevA.100.022512. URL https://link.aps.org/doi/10.1103/PhysRevA.100.022512.

[13] K. T. Schütt, M. Gastegger, A. Tkatchenko, K.-R. Müller, and R. J. Maurer. Unifying machine learning and quantum chemistry with a deep neural network for molecular wavefunctions. *Nature Communications*, 10(1):5024, Nov 2019. ISSN 2041-1723. doi: 10.1038/s41467-019-12875-2. URL https://doi.org/10.1038/s41467-019-12875-2.

[14] Bodille Blomaard, Lucas Crijns, Arun Karim, and Thomas Pijls. Dftb machine learning project report. De Boelelaan 1083, Amsterdam, 2018. URL https://www.scm.com/about-us/. High school project executed at Software for Chemistry & Materials BV.

[15] D. J. Griffiths. *Introduction to Quantum Mechanics*. Pearson Education (Us), 3 edition, 2018.

[16] 8.1: Atomic and molecular calculations are expressed in atomic units. https://chem.libretexts.org/Courses/Pacific_Union_College/Quantum_Chemistry/08%3A_Multielectron_Atoms/8.01%3A_Atomic_and_Molecular_Calculations_are_Expressed_in_Atomic_Units#:~:text=Atomic%20units%20(au%20or%20a.u.,of%20length%2C%20and%20so%20on., 2020. Accessed: 4 August 08 2020.

[17] Klaas Giesbertz. Hartree-fock & density functional theory. Understanding quantum chemistry lecture notes, De boelelaan 1105, Amsterdam, 2019. URL http://few.vu.nl/~kjgiesbe/. Universiteit van Amsterdam & Vrije Universiteit Amsterdam.

[18] P. Atkins and R. Friedman. *Molecular Quantum Mechanics*. Oxford, 5 edition, 2010.

[19] C. C. J. Roothaan. New developments in molecular orbital theory. *Rev. Mod. Phys.*, 23:69–89, Apr 1951. doi: 10.1103/RevModPhys.23.69. URL https://link.aps.org/doi/10.1103/RevModPhys.23.69.

[20] P. Hohenberg and W. Kohn. Inhomogeneous electron gas. *Phys. Rev.*, 136:B864–B871, Nov 1964. doi: 10.1103/PhysRev.136.B864. URL https://link.aps.org/doi/10.1103/PhysRev.136.B864.

[21] Carlo Adamo and Vincenzo Barone. Physically motivated density functionals with improved performances: The modified perdew–burke–ernzerhof model. *The Journal of Chemical Physics*, 116(14):5933–5940, 2002. doi: 10.1063/1.1458927. URL https://doi.org/10.1063/1.1458927.

[22] Pekka Koskinen and Ville Mäkinen. Density-functional tight-binding for beginners. *Computational Materials Science*, 47(1):237–253, Nov 2009. ISSN 0927-0256. doi: 10.1016/j.commatsci.2009.07.013. URL http://dx.doi.org/10.1016/j.commatsci.2009.07.013.

[23] R. Rüger. *Approximations in Density Functional Based Excited State Calculations*. PhD thesis, Vrije Universiteit Amsterdam, 2018.

[24] Mulliken, R. S. Quelques aspects de la théorie des orbitales moléculaires. *J. Chim. Phys.*, 46:497–542, 1949. doi: 10.1051/jcp/1949460497. URL https://doi.org/10.1051/jcp/1949460497.

[25] Dulal C. Ghosh and Nazmul Islam. Semiempirical evaluation of the global hardness of the atoms of 103 elements of the periodic table using the most probable radii as their size descriptors. *International Journal of Quantum Chemistry*, 110(6):1206–1213, 2010. doi: 10.1002/qua.22202. URL https://onlinelibrary.wiley.com/doi/abs/10.1002/qua.22202.

[26] Michael Gaus, Qiang Cui, and Marcus Elstner. Dftb3: Extension of the self-consistent-charge density-functional tight-binding method (scc-dftb). *Journal of Chemical Theory and Computation*, 7(4):931–948, 2011. doi: 10.1021/ct100684s. URL https://doi.org/10.1021/ct100684s.

[27] Charu C. Aggarwal. *An Introduction to Neural Networks*, pages 1–52. Springer International Publishing, Cham, 2018. ISBN 978-3-319-94463-0. doi: 10.1007/978-3-319-94463-0_1. URL https://doi.org/10.1007/978-3-319-94463-0_1.

[28] Christopher M. Bishop. *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer, 1 edition, 2006.

[29] Jason Brownlee. A gentle introduction to the rectified linear unit (relu). https://machinelearningmastery.com/rectified-linear-activation-function-for-deep-learning-neural-networks/, 2020. Accessed: 30 July 2020.

[30] Jason Brownlee. How to use data scaling improve deep learning model stability and performance. https://machinelearningmastery.com/how-to-improve-neural-network-stability-and-modeling-performance-with-data-scaling/, 2019. Accessed: 31 July 2020.

[31] Jason Brownlee. A gentle introduction to mini-batch gradient descent and how to configure batch size. https://machinelearningmastery.com/gentle-introduction-mini-batch-gradient-descent-configure-batch-size/, 2019. Accessed: 31 July 2020.

[32] Sanket Doshi. Various optimization algorithms for training neural network: The right optimization algorithm can reduce training time exponentially. https://medium.com/@sdoshi579/optimizers-for-training-neural-network-59450d71caf6, 2019. Accessed: 31 July 2020.

[33] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization, 2014.

[34] Kurtis Pykes. Adam optimization algorithm: Overview of an effective optimization algorithm. https://towardsdatascience.com/adam-optimization-algorithm-1cdc9b12724a, 2020. Accessed: 31 July 2020.

[35] Tara Boyle. Hyperparameter tuning: Exploring hyperparameter tuning methods in kaggle's don't overfit ii competition. https://towardsdatascience.com/hyperparameter-tuning-c5619e7e6624, 2019. Accessed: 14 August 2020.

[36] Jason Brownlee. Use early stopping to halt the training of neural networks at the right time. https://machinelearningmastery.com/how-to-stop-training-deep-neural-networks-at-the-right-time-using-early-stopping/, 2019. Accessed: 31 July 2020.

[37] Jason Brownlee. Use weight regularization to reduce overfitting of deep learning models. https://machinelearningmastery.com/weight-regularization-

to-reduce-overfitting-of-deep-learning-models/, 2019. Accessed: 31 July 2020.

[38] Jason Brownlee. A gentle introduction to dropout for regularizing deep neural networks. https://machinelearningmastery.com/dropout-for-regularizing-deep-neural-networks/, 2020. Accessed: 31 July 2020.

[39] John P. Perdew, Kieron Burke, and Matthias Ernzerhof. Generalized gradient approximation made simple. *Phys. Rev. Lett.*, 77:3865–3868, Oct 1996. doi: 10.1103/PhysRevLett.77.3865. URL https://link.aps.org/doi/10.1103/PhysRevLett.77.3865.

[40] Matthias Ernzerhof and Gustavo E. Scuseria. Assessment of the perdew–burke–ernzerhof exchange-correlation functional. *The Journal of Chemical Physics*, 110(11):5029–5036, 1999. doi: 10.1063/1.478401. URL https://doi.org/10.1063/1.478401.

[41] Susi Lehtola. Assessment of initial guesses for self-consistent field calculations. superposition of atomic potentials: Simple yet efficient. *Journal of Chemical Theory and Computation*, 15(3):1593–1604, 2019. doi: 10.1021/acs.jctc.8b01089. URL https://doi.org/10.1021/acs.jctc.8b01089. PMID: 30653322.

[42] Jason Brownlee. Why one-hot encode data in machine learning? https://machinelearningmastery.com/why-one-hot-encode-data-in-machine-learning/, 2020. Accessed: 11 August 2020.

[43] Qiming Sun, Timothy C. Berkelbach, Nick S. Blunt, George H. Booth, Sheng Guo, Zhendong Li, Junzi Liu, James D. McClain, Elvira R. Sayfutyarova, Sandeep Sharma, Sebastian Wouters, and Garnet Kin-Lic Chan. Pyscf: the python-based simulations of chemistry framework, 2017. URL https://onlinelibrary.wiley.com/doi/abs/10.1002/wcms.1340.

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

[45] Christoph Schober, Karsten Reuter, and Harald Oberhofer. Critical analysis of fragment-orbital dft schemes for the calculation of electronic coupling values. *The Journal of Chemical Physics*, 144(5):054103, 2016. doi: 10.1063/1.4940920. URL https://doi.org/10.1063/1.4940920.

[46] Jörg Behler. Atom-centered symmetry functions for constructing high-dimensional neural network potentials. *The Journal of Chemical Physics*, 134(7):074106, 2011. doi: 10.1063/1.3553717. URL https://doi.org/10.1063/1.3553717.

[47] G. te Velde, F. M. Bickelhaupt, E. J. Baerends, C. Fonseca Guerra, S. J. A. van Gisbergen, J. G. Snijders, and T. Ziegler. Chemistry with adf. *J. Comput. Chem.*, 22(9):931–967, 2001. ISSN 1096-987X. doi: 10.1002/jcc.1056. URL http://dx.doi.org/10.1002/jcc.1056.

[48] The dftb website. URL http://www.dftb.org/. Accessed: 31 July 2020.

[49] Lauri Himanen, Marc O. J. Jäger, Eiaki V. Morooka, Filippo Federici Canova, Yashasvi S. Ranawat, David Z. Gao, Patrick Rinke, and Adam S. Foster. DScribe: Library of descriptors for machine learning in materials science. *Computer Physics Communications*, 247:106949, 2020. ISSN 0010-4655. doi: 10.1016/j.cpc.2019.106949. URL https://doi.org/10.1016/j.cpc.2019.106949.

[50] 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.*, 108:058301, Jan 2012. doi: 10.1103/PhysRevLett.108.058301. URL https://link.aps.org/doi/10.1103/PhysRevLett.108.058301.

[51] Datasets: A qm/ml resource. https://qmml.org/datasets.html. Accessed: 25 July 2020.

[52] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016. http://www.deeplearningbook.org.

[53] D R Yarkony. *Modern Electronic Structure Theory*. World Scientific Publishing Company, 1995. doi: 10.1142/1957. URL https://www.worldscientific.com/doi/abs/10.1142/1957.

[54] Raymond P. Iczkowski and John L. Margrave. Electronegativity. *Journal of the American Chemical Society*, 83(17):3547–3551, 1961. doi: 10.1021/ja01478a001. URL https://doi.org/10.1021/ja01478a001.

[55] Mark S. Gordon, J. Stephen Binkley, John A. Pople, William J. Pietro, and Warren J. Hehre. Self-consistent molecular-orbital methods. 22. small split-valence basis sets for second-row elements. *Journal of the American Chemical Society*, 104 (10):2797–2803, 1982. doi: 10.1021/ja00374a017. URL https://doi.org/10.1021/ja00374a017.

Appendix

A. Full configuration-interaction & Post Hartree-Fock methods

It turns out that a single SD is not enough to describe the ground state wavefunction. This is where full configuration-interaction (CI) comes in. This methods yields the exact eigenstates of the molecular Hamiltonian within a given basis. Now to find the exact eigenstates of the molecular Hamiltonian we first note that electrons are indistinguishable. This means that $\binom{M}{N}$ unique SDs can be defined for a system with N electrons described by M one-electron basis functions. Linear combinations of these SDs can now be used to represent the eigenstates of the molecular Hamiltonian. Similarly to HF, the Hamiltonian can be discretized as

$$\mathrm{H}_{ij}^{mol} = \langle \psi_i^{SD} | \hat{H}^{mol} | \psi_j^{SD} \rangle \tag{8.1}$$

which is an $\binom{M}{N} \times \binom{M}{N}$ matrix. The discretized eigenvalue equation now becomes

$$\mathbf{H}^{mol}\mathbf{C} = \mathbf{C}\mathrm{diag}(\mathbf{E}) \tag{8.2}$$

from which we obtain the eigenstates as linear combination of SDs described by the columns of $\mathbf{C}$, along with corresponding eigen energies $\{E_i\}$. However, the major downside to this method is that the amount of operations required to solve the equations scales factorially with the number of electrons and the size of the basis set [4, 17].

Since full configuration-interaction (CI) scales horribly with increasing system size there is a need for better scaling methods that add some of the electron correlation effects, not present in HF, to the wavefunction. For this purpose there are several post-HF methods available including truncated CI, coupled-cluster (CC), and Møller–Plesset perturbation theory (MP). Truncated CI differs from full CI in the fact it starts off with the HF ground state SD and adds a subset of excited SDs. To construct these additional SDs it uses the unoccupied higher energy MOs obtained during the HF procedure, *e.g* only singly

(CIS) or singly and doubly excited (CISD) ones. The eigenstates are once again obtained through solving equation 8.2. CC takes a similar approach with the main difference being that it implicitly takes all possible excitations into account. An advantage of this is that, contrary to CI, the energy of infinitely dissociated atoms equals the sum of the energy of each individual atom in vacuum, as it should. Again CC comes in many flavours such as coupled-cluster singles (CCS) and coupled-cluster singles-doubles (CCSD). MP is a different "beast" which improves upon the HF results by accounting for the difference between the Fock operator and the true molecular Hamiltonian by expanding in orders of correction [4, 17].

B. Basis functions

The two types of basis functions that are most frequently employed for electronic structure calculations are Slater-type orbitals (STOs) and Gaussian-type orbitals (GTOs). STOs more accurately describe the behaviour of electrons, however in practice GTOs are usually preferred, since these are less costly to integrate. The downside to this it that a single Gaussian function behaves according to $\frac{1}{r^2}$ when moving away from the nucleus, in contrast to an STO which decreases as $\frac{1}{r}$. Another advantage of STOs is that they correctly replicate the cusp of the wavefunction at a nucleus. In order to combat these deficiencies one typically resorts to a least-squares fit of n Gaussians to mimic a single STO. Combining this with a minimal basis set, *i.e.* the minimum number of orbitals required to accommodate the number of electrons present in the atom, is referred to as STO-nG [18].