Klemens Schmuck BSc.

Transition state search accelerated with machine learning techniques

MASTER'S THESIS

to achieve the university degree of

Diplom-Ingenieur

Master's degree programme: Technical Physics

submitted to

Graz University of Technology

Supervisor

Ass.Prof. Mag.phil. Dipl.-Ing. Dr.phil. Dr.techn. Andreas W. Hauser

Co-supervisor
Dipl. Ing. Ralf Meyer

Institut für Experimentalphysik

Graz, March 2019

AFFIDAVIT

I declare that I have authored this thesis independently, that I have not used other than the declared sources/resources, and that I have explicitly indicated all material which has been quoted either literally or by content from the sources used. The text document uploaded to TUGRAZonline is identical to the present master‘s thesis.

Date

Signature

Abstract

Accurate knowledge of reactions is essential in chemistry. Thus, calculations of the reaction path, which is defined as the minimum energy pathway that connects two minima, are crucial. On this path the metastable transition state is of special interest, since it defines the chemical reactivity and distribution of reaction products. The nudged elastic band method allows to locate this extremum of the potential energy surface, but since this technique typically requires many *ab initio* calculations, it is computationally rather expensive, in particular for larger molecular systems.

Machine learning techniques are able to learn high dimensional scalar functions, which makes them suitable tools to accelerate the transition state search. In order to do so, the nudged elastic band method is applied to an approximated potential energy surface created and gradually improved by the machine learning algorithm.

In this thesis the following four machine learning techniques are used: regularised least squares, Gaussian process regression, iterative re-weighted least squares and a neural network-based approach. All these methods are tested on well known benchmark systems and compounds regarding to their predictions of the transition state and the number of *ab initio* calls needed for convergence. Two fundamental structural changes in molecules have been investigated, namely inversion and rotation.

Zusammenfassung

In der Chemie ist eine detaillierte Kenntnis von Reaktionsprozessen und deren Teilschritte von größter Bedeutung. Diese werden durch den Reaktionspfad, welcher als Pfad der kleinsten Energie zwei Minima auf der molekularen Potentialenergieoberfläche verbindet, definiert. Auf diesem Pfad liegt auch der metastabile Sattelpunkt, der sowohl die chemische Reaktivität als auch die Verteilung möglicher Produkte bestimmt und dadurch von besonderem Interesse ist. Mit Hilfe der "Nudged-Elastic-Band" Methode ist es möglich, den Reaktionspfad und die Sattelpunktsgeometrie zu bestimmen; allerdings ist dieser Ansatz für größere Moleküle sehr zeitintensiv.

Um diese Suche zu beschleunigen, kommt in dieser Arbeit maschinelles Lernen als Konzept zum Einsatz, da derartige Verfahren in der Lage sind, auch hochdimensionale skalare Funktionen nachzubilden. Dabei wird die "Nudged-Elastic-Band" Methode auf der fiktiven Oberfläche, die durch den Lernalgorithmus erzeugt wird, angewandt, während die Oberfläche selbst in jedem Schritt weiter verbessert wird.

Vier unterschiedliche Lernmethoden werden vergleichend gegenübergestellt und näher analysiert: "Regularised Least Squares", "Gaussian Process Regression", "Iterative Re-Weighted Least Squares" und ein "Neural Network". Diese Methoden werden auf Molekülsystemen getestet und in Bezug auf deren Genauigkeit anhand des lokalisierten Sattelpunktes und der Anzahl an nötigen *ab initio*-Aufrufen verglichen. Die Auswahl an molekularen Systemen trägt dem Versuch Rechnung, typische Änderungen der Geometrie, wie sie bei Reaktionen auftreten können, an möglichst kleinen, jedoch gut bekannten Testmolekülen nachzubilden; diese umfassen Inversion und Rotation.

Acknowledgement

I would like to thank my supervisor Prof. Andreas Hauser for giving me the opportunity to write this thesis in his group and for his quick response to all my concerns during this time. Furthermore, I appreciate that I had the chance to attend an international conference. I would also like to thank my colleges and my co-supervisor Ralf Meyer for his continuous support and for his patient guidance throughout this thesis.

I would like to thank my family, especially my parents, who supported me financially and mentally during my studies. Further, I am deeply grateful to Kathrin for her support and for pulling me out of my working routines when I really needed a break.

At last, I want to express my sincere gratitude to all my friends for the great times we had as students, thank them for their motivation and the gentle push I needed here and there to finish my work.

Contents

List of Figures

List of Tables

1. Introduction

The chemical behaviour of molecules is completely described by the potential energy surface, which is given as high-dimensional scalar function of the nuclear coordinates. This landscape is like a fingerprint, and the knowledge of its shape is crucial in chemistry. Nowadays, well established *ab initio* methods, such as wave-function based methods or density functional theory, allow to explore this surface to good accuracy, but these calculations are time-consuming, especially for larger systems.

The objective function of these calculations is the Schrödinger equation. For problems discussed in this thesis, it is sufficient to use the time-independent Schrödinger equation, which is given by

$$\mathcal{H}_{\text{molecular}} \left|\psi(\boldsymbol{R}, \boldsymbol{r})\right\rangle = E \left|\psi(\boldsymbol{R}, \boldsymbol{r})\right\rangle , \tag{1.1}$$

with E as the eigenvalue, $|\psi(\boldsymbol{R}, \boldsymbol{r})\rangle$ as eigenfunctions and $\mathcal{H}_{\text{molecular}}$ as the molecular Hamiltonian, which is given in atomic units by

$$\begin{aligned} \mathcal{H}_{\text{molecular}} = &- \sum_{i=1}^{N_{\text{elec}}} \frac{1}{2} \boldsymbol{\nabla}_i^2 - \sum_{A=1}^{N_{\text{nuclei}}} \frac{1}{2M_A} \boldsymbol{\nabla}_A^2 - \sum_{i=1}^{N_{\text{elec}}} \sum_{A=1}^{N_{\text{nuclei}}} \frac{Z_A}{|\boldsymbol{R}_A - \boldsymbol{r}_i|} \\ &+ \sum_{i=1}^{N_{\text{elec}}} \sum_{j>i}^{N_{\text{elec}}} \frac{1}{|\boldsymbol{r}_i - \boldsymbol{r}_j|} + \sum_{A=1}^{N_{\text{nuclei}}} \sum_{B>A}^{N_{\text{nuclei}}} \frac{Z_A Z_B}{|\boldsymbol{R}_A - \boldsymbol{R}_B|}, \end{aligned} \tag{1.2}$$

with the upper case letters denoting nuclei and the lower case ones denoting electrons. This Hamiltonian describes a many-body interaction; therefore, the Schrödinger equation can not be solved directly, but several approximations have to be employed.

The first simplification we apply to the Schrödinger equation of the total molecular system is the Born-Oppenheimer approximation, which assumes that it is possible to decouple the motion of electrons and nuclei. It is assumed that the nuclei are at rest and only the electrons move. The justification for that is given by the mass ratio of electron and nuclei $\frac{m_{\text{nuclei}}}{\text{m}_{\text{elec}}} \approx 1838$. In this approximation, the position variables of the nuclei turn into parameters. The electronic Hamiltonian to be solved can then be written as

$$\begin{aligned} \mathcal{H}_{\text{elec}} = &- \sum_{i=1}^{N_{\text{elec}}} \frac{1}{2} \boldsymbol{\nabla}_i^2 - \sum_{i=1}^{N_{\text{elec}}} \sum_{A=1}^{N_{\text{nuclei}}} \frac{Z_A}{|\boldsymbol{R}_A - \boldsymbol{r}_i|} \\ &+ \sum_{i=1}^{N_{\text{elec}}} \sum_{j>i}^{N_{\text{elec}}} \frac{1}{|\boldsymbol{r}_i - \boldsymbol{r}_j|} + \sum_{A=1}^{N_{\text{nuclei}}} \sum_{B>A}^{N_{\text{nuclei}}} \frac{Z_A Z_B}{|\boldsymbol{R}_A - \boldsymbol{R}_B|}. \end{aligned} \tag{1.3}$$

and the total wave function is a product of electron and nuclei wave function $\psi(\boldsymbol{R}, \boldsymbol{r}) = \xi(\boldsymbol{R})\Phi(\boldsymbol{R}, \boldsymbol{r})$. This separates the Schrödinger equation into a nuclear and electronic part. The latter depends only on the nuclear positions as parameters and it is obtained by replacing the molecular Hamiltonian in Equation 1.1 with the electronic Hamiltonian. Solving this equation yields the total electronic wave function and its corresponding energy as a function of nuclear coordinates, which defines the potential energy surface. Since the nuclei position are used as parameter it is possible to explore this landscape by adapting the nuclei positions.

Knowledge about this potential energy surface also allows to simulate reactions. The latter are described by the minimum energy path which connects a reactant with a product state. Reaction rates and selectivities are completely defined by the meta stable state on their minimum energy path, the transition state, see Section 1.3. This special point determines the energy barrier between the two minima. It is possible to manipulate the transition state with catalysts, which are additional substances that are present during the reaction but are not consumed by it. These substances may affect the barrier height and make reactions possible, faster or less energy intensive, but they can also stop reactions as well. Experimentally, it is exhausting and laborious to search for such substances by trial and error. Hence, computer experiments are a welcome tool to reduce the effort. For reactions, these simulations focus on the transition state search, which requires many single point *ab initio* calculations, making this search computationally expensive.

The transition state search can be accelerated by machine learning algorithms. Several different methods have been proposed to accelerate the procedure [1][2][3]. In this thesis, an overview of selected approaches is given and the support vector machine formalism is added. All methods are finally applied to two well known molecular reactions and are compared with respect to the number of *ab initio* calculations.

1.1. Hartree-Fock (HF)

The Hartree-Fock method simplifies the Born-Oppenheimer approximation further by assuming that the total electronic wave function can be written as a product of molecular orbitals. Since electrons are fermions, a Slater determinate is used for that purpose It ensures that the total wave function is antisymmetric and does not violate Pauli's principle. This allows to write the Hamiltonian as a product of one-electron Hamiltonians and leads in the course of applying the variational principle to the Hartree-Fock equations given by

$$\begin{aligned} f(1)\chi_a(1) = \epsilon_a\chi_a(1) = h(1)\chi_a(1) + \sum_{a\neq b} \underbrace{\left[\int dx_2 \frac{|\chi_b(2)|^2}{r_{1,2}}\right]}_{J} \chi_a(1) \\ - \sum_{a\neq b} \underbrace{\left[\int dx_2 \frac{\chi_b^*(2)\chi_a(2)}{r_{1,2}}\right]}_{K} \chi_b(1), \end{aligned} \tag{1.4}$$

where the subscripts a and b are the orbital index, χ is the spin-orbital function, ϵ is the eigen energy of the electron and h is the one electron Hamiltonian. In this formula, $f(1)$ denotes the Fock operator, an effective one electron operator, on the first electron in spin orbital χ_a. The Coulomb integral J and the exchange integral K describe the interaction between the electrons, where one electron interacts with the average charge distribution of all other electrons. Hence, the solution of this equation depends on the solution of all the other electrons in the system. This necessitates the Hartree-Fock equations to be solved iteratively. Solutions of the Hartree-Fock equations are obtained by minimising the energy with respect to the molecular orbitals. In order to solve it, a basis expansion of the molecular orbitals is introduced, typically. A convenient basis are the atomic orbitals due to their asymptotic correctness at large distances. This expansion is given by

$$\chi_a = \sum_{\mu=1}^{K} c_{\mu a}\phi_\mu, \tag{1.5}$$

where ϕ are the atomic orbitals, c are coefficients to the orbitals and K is the number of included orbitals. The atomic orbitals are kept fixed. Therefore, the two-electron integrals have only to be evaluated once. In order to minimise the energy only the coefficients have to be varied. This allows to rewrite the Hartree-Fock equations in an algebraic form. In matrix notation, one obtains,

$$FC = SC\epsilon \tag{1.6}$$

with F as the Fock matrix, C as the coefficient matrix, S as the overlap matrix and ϵ as diagonal matrix containing the energies. The solution of the coefficient matrix can be found by diagonalising the Fock matrix. However, the Hartree-Fock method only approximates the ground state of a molecule within the mean field approximation and even in the limit of an infinite number of atomic orbitals it does not reach the exact solution, due to the lack of correlation energy. The interested reader is referred to the book of Szabo et al. [1].

1.2. Density Functional Theory (DFT)

Density functional theory is based on the two theorems of Hohenberg and Kohn. The first states that two different external potentials will not give the same electronic density, as it is proven in Appendix A.2. The second theorem states that the exact ground state is the global minimum of the universal energy functional with respect to the electron density. In other words, the density that minimises the energy is the correct ground state density. With these two theorems, density functional theory can be stated as a minimisation of the energy as a functional of the density, depending on the number of electrons in the system and the actual molecular geometry within the Born-Oppenheimer approximation.

In order to solve the Schrödinger equation, the Hamiltonian has to be rewritten in terms of the density. The Coulomb part and the nuclei electron interaction are the easy parts, which can be simply written as

$$E_{\mathrm{ne}}(\rho) = -\sum_{A}^{N_{\mathrm{nuclei}}} \int \mathrm{d}\boldsymbol{r} \frac{Z_A \boldsymbol{R}_A \rho(\boldsymbol{r})}{|\boldsymbol{R}_A - \boldsymbol{r}|}, \tag{1.7}$$

$$J(\rho) = \frac{1}{2} \int\int \mathrm{d}\boldsymbol{r}\mathrm{d}\boldsymbol{r}' \frac{\rho(\boldsymbol{r})\rho(\boldsymbol{r}')}{|\boldsymbol{r} - \boldsymbol{r}'|}, \tag{1.8}$$

with $E_{\mathrm{ne}}(\rho)$ as the nuclei electron interaction and $J(\rho)$ as Coulomb part. Unfortunately, the kinetic energy and the exchange part cannot be expressed in an analytical fashion and have to be approximated. A common way to estimate the kinetic part is to reintroduce orbitals to the formalism. This leads to the Kohn-Sham density functional theory, where the kinetic energy is separated into two parts. The larger part describes the non-interacting kinetic energy which can be accurately calculated with the orbitals. The smaller part remains unknown and is added to the exchange part to give the exchange correlation energy, which has to be guessed.

$$E_{\mathrm{DFT}}(\rho) = T_{\mathrm{s}}(\rho) + E_{\mathrm{ne}}(\rho) + J(\rho) + E_{\mathrm{xc}}(\rho) \tag{1.9}$$

[1] A full description of Hartree-Fock can be found in the book Szabo et al. [4], page 108 et sequentes.

$$E_{\mathrm{xc}}(\rho) = \underbrace{(T(\rho) - T_{\mathrm{s}}(\rho))}_{\text{kinetic correlation}} + \underbrace{(E_{\mathrm{ee}}(\rho) - J(\rho))}_{\text{exchange-correlation}}. \tag{1.10}$$

In this equation, the exchange correlation functional $E_{\mathrm{xc}}(\rho)$ collects all unknown contributions; its actual form remains yet unknown. For further details to density functional theory refer to the book of Burke [2] and the book of Paar [3].

1.3. Transition state search

The transition state is of special interest for chemists, because it allows to calculate the distribution of reactants and chemical reactivities. This state corresponds to the configuration of highest energy on the minimum energy path and is metastable. There are several well-established methods available, such as eigenvector following techniques [7], growing or freezing string methods [8][9][10] and nudged elastic band calculations [11], which allow a localisation of the transition state. This thesis focuses on the nudged elastic band algorithm, which has the advantage of approximating the minimum energy path as well as locating the transition state.

The estimation of the minimum energy path requires two minima on the landscape. At the beginning, these two are connected via a linearly interpolated path consisting of N images. Each of them describes a new position on the potential energy surface. Neighbouring images are connected with a fictitious spring to keep the images equidistantly distributed along the path [12][13].

The basic formulation minimises the energy objective function, which is a sum over the spring and landscape energy. However, this may lead to corner-cutting and sliding-down issues, because the spring force and the negative landscape gradient are able to have the same direction. Thus, they might cancel each other. Corner-cutting occurs if the connecting spring is hard and the path is kinky, so that the image can not follow the negative gradient. On the other hand if the spring is too soft, it does not preserve equidistance between images. Also, the sliding-down issue may occur.
The problems are solved by projecting the spring forces into a parallel direction to the band, while forces acting due to the surface are only allowed to act perpendicular to the band. However, this does not prevent the band from overall stretching. In this case, a reaction path of overestimated length is assumed, which might be energetically but not entropically favourable [13]. The total force acting on each image is then given by

$$\boldsymbol{F}_i = -[\nabla_i E(\boldsymbol{x}_i) - ([\nabla_i E(\boldsymbol{x}_i)]\hat{\boldsymbol{\tau}}_i)\hat{\boldsymbol{\tau}}_i] + (\boldsymbol{F}_i^{\mathrm{s}}\hat{\boldsymbol{\tau}}_i)\hat{\boldsymbol{\tau}}_i, \tag{1.11}$$

[2] Fundamental explanation to density functional theory can be found in [5].
[3] [6] gives a detailed description of density functional theory.

with $\hat{\boldsymbol{\tau}}_i$ as tangent to the band, $\boldsymbol{x}_i$ as position, $\boldsymbol{F}_i^{\mathrm{s}}$ as spring force and the subscript i denoting the image. The spring force can be expressed as

$$F_i^{\mathrm{s}} = k[(\boldsymbol{x}_{i+1} - \boldsymbol{x}_i) - (\boldsymbol{x}_i - \boldsymbol{x}_{i-1})], \tag{1.12}$$

where k is the spring constant. Since the band in general has no smooth curvature, the tangent is estimated. In the simplest case it is sufficient to calculate the tangent mathematically with respect to the neighbouring images [13],

$$\hat{\boldsymbol{\tau}}_i = \frac{\boldsymbol{x}_{i+1} - \boldsymbol{x}_{i-1}}{|\boldsymbol{x}_{i+1} - \boldsymbol{x}_{i-1}|}. \tag{1.13}$$

Improvements and different estimates to approximate the tangent can be found in Appendix A.3.
The main procedure of the nudged elastic band comprises the following steps:

1. Initialise the band with a linear interpolation between the two minima.
2. Evaluate the energy and gradient for each image on the potential energy surface.
3. Estimate every tangent, and calculate the spring and acting force of each image.
4. Move every image on the potential energy surface with respect to the acting force.
5. Continue with step 2 until the acting force norms are below a certain threshold and convergence is achieved.

The nudged elastic band algorithm convergences towards the minimum energy path, but it does not deliver the transition state automatically. This feature can be added by first converging the algorithm to the minimum energy path and then applying the climbing image method, where the spring force of the image with highest energy is set to zero. As a consequence, in contrast to the standard algorithm, the image with highest energy has to follow the positive potential energy gradient without changing the sign and moves therefore towards the saddle point [13].

The initial linear interpolation of the band is sometimes problematic because atoms can get too close to each other or may even collide upon rearrangement. This issue can be solved with the so called image-dependent pair potential technique [14], where the real landscape is replaced with a well-behaved, analytical surface that is cheap to evaluate. For that, the algorithm first calculates the pairwise distance between all atoms at each minimum geometry. These distances are then used to linearly interpolate the pairwise distance for each intermediate image, which gives the optimal distance between the

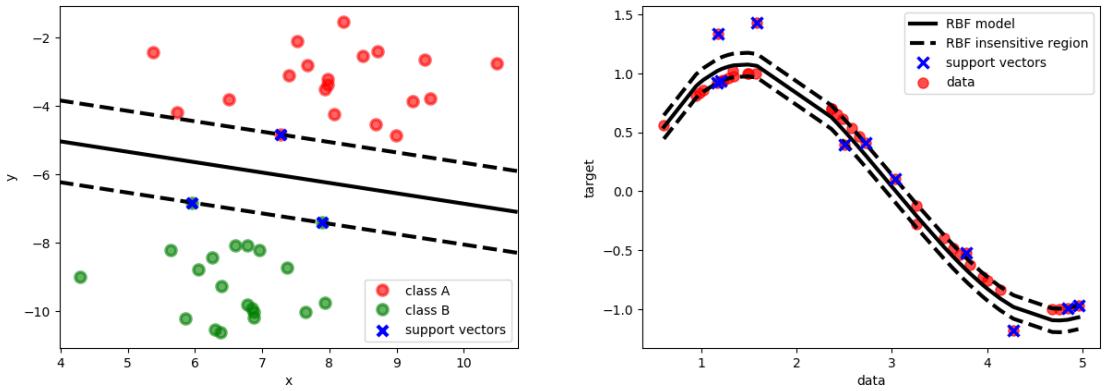
atoms. For each atom the deviation of the actual pairwise distances to the optimal is calculated. The image-dependent pair potential is given as a sum over these deviations multiplied with a weight to penalise small pairwise distances. The nudged elastic band method can then be applied to this potential in order to estimate an initial pathway on which atom collisions are avoided.

With this connection of the first estimation of the pathway, an unphysical proximity of atoms of intermediate images is avoided, and the nudged elastic band algorithm can be applied to the real potential energy surface after. For further details on the image-dependent pair potential see Appendix A.4.

2. Machine Learning

In the last decade machine learning has gained a lot of popularity, due to the tremendous increase of computer power. In machine learning, two different learning concepts can be distinguished. One is the unsupervised learning; in this case no information about how the prediction should look like is available. The other method is supervised learning, where data points with the corresponding value are known. The focus of this thesis will be on the supervised learning, because information on output values, in our case the electronic energies, exists. For more details on unsupervised learning see the book of Duda et al. [1].

The problems of supervised learning can be separated in two main classes: classification and regression. Classification accepts only discrete labels, which are the corresponding output values to the input data. This approach separates and matches data to specific classes. On the other hand, regression accepts all possible labels or output values, and fits the data points accordingly. A simple example is shown in Figure 2.1. This thesis focuses on the prediction of molecular energies on a potential energy surface to reduce the amount of *ab initio* calculations. Therefore, it will be concerned with regression problems only. Further details to classification can be found in the book by Duda et al. [2].

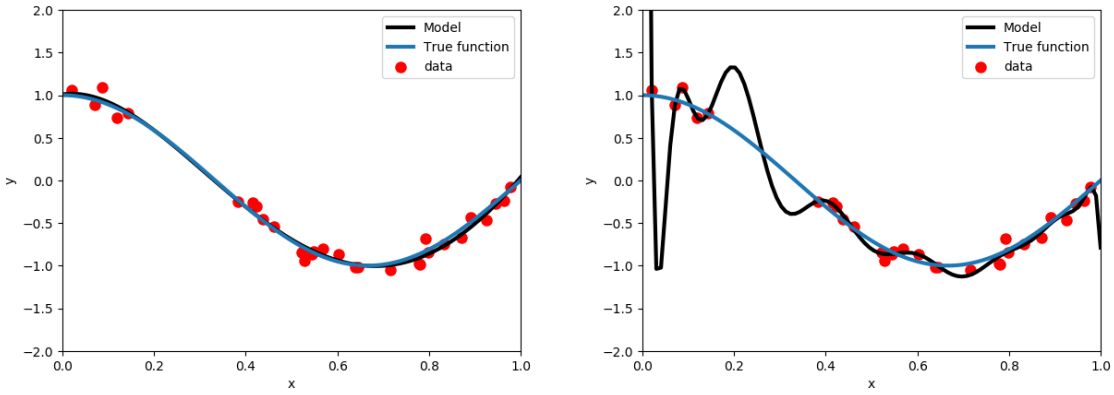


(a) Classification [16] (b) Regression [17]

Figure 2.1.: A simple example for the two different supervised machine learning problems, solved with a support vector machine. In both cases the black line represents the fitted plane and the two dotted black lines are the decision boundaries.

[1] Details to unsupervised learning can be found in [15], page 603 et sequentes.
[2] An introduction to classification is given in [15] page 9 et sequentes.

The goal of regression algorithms is to represent training data as well as the underlying function connecting input and output. This is achieved by minimising the prediction error on the training data. Reducing the error to zero would represent the training data best, but might indicate a situation of overfitting. In that case unseen data points are not predicted correctly and the machine learning algorithm produces a wrong representation of the underlying function, see Figure 2.2. To avoid this, a smooth function curvature is preferred in the training process. Thus, a weight regularisation is introduced.



(a) well fitted curve (b) over fitted curve

Figure 2.2.: Simple example for overfitting behaviour taken from the scikit-learn website [18]

Training Data

In the following, $\boldsymbol{x}$ denotes a vector with m features and $y(\boldsymbol{x})$ is the corresponding function value with dimension one. The input for training the algorithm is rewritten in matrix form as X with the dimension $[N, m]$, which contains N row vectors $\boldsymbol{x}$, and the output is a column vector $\boldsymbol{y}(X)$ with N values. N is defined as the number of samples and m represents the dimension of each sample.

2.1. Neural Network (NN)

The neural network model is inspired by the human brain. It is able to learn any kind of continuous and smooth function mapping a high dimensional input to a scalar value. In principle, the network consists at least of two layers: the input and output layer. Additional layers between them are called hidden layers. Each layer contains nodes that are equivalent to neurons in the brain. Every layer can have a different number of nodes except for the output, layer which always has just one, and the input layer which has to have at least the number of features as nodes. However, note that the number

of nodes can only be changed before the network is trained. Each node in one layer is connected with all nodes of the previous layer. Figure 2.3 shows the architecture of a neural network.

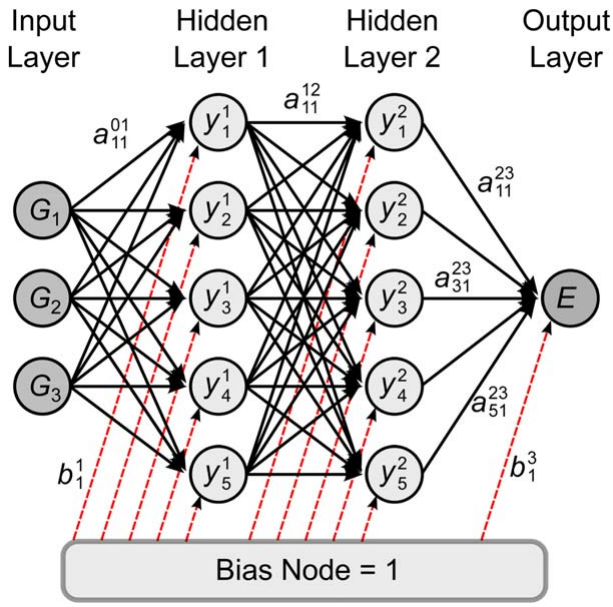


Figure 2.3.: Architecture of a neural network with two hidden layer. The picture is taken from Behler et al. [19].

Every node applies a weight to the value of each connection of the previous layer, adds a bias vector to it and sums over all connections. This is then fed to an activation function $f_a(x)$ before propagating the value to the next layer. Doing this for all hidden layers and the output layer leads to a scalar output value. The evaluation function of a network with one hidden layer is given by

$$E = f_a(\boldsymbol{x}A_0 + \boldsymbol{b}_0)A_1 + b_1, \tag{2.1}$$

with E as the scalar output value, $\boldsymbol{x}$ as input vector, A_i as weight matrix, $\boldsymbol{b}_i$ as bias and i as the layer index. Note that, for the output layer the weight matrix is a vector and the bias is scalar, since only one neuron is present. In case of regression the output layer consists of a linear activation function. The training process adjusts the weights and biases by lowering the output error. This error is calculated by the loss function which measures the difference between output value and input function value. After training the network, it is tested with validation data which has not been used in the training process. This allows to detect and prevent from over fitting, because if the validation error remains too high the training is considered as failure.

Neural networks have the advantage that they do not scale with the amount of input data, but in order to adjust all weights correctly in the network a large amount of training data is needed. A disadvantage is that the process is sensitive to feature scaling and that the loss function is not convex. Due to this many local minima exist and it depends on the weight initialisation which minimum is obtained eventually.

2.2. Kernel-Based Methods

The basic algorithms of kernel-based machine learning methods use a linear function to fit data [20][21], which is given as

$$f(\boldsymbol{x}_{\text{pred}}) = \widetilde{\boldsymbol{w}}\boldsymbol{x}_{\text{pred}} + b, \tag{2.2}$$

where $\boldsymbol{x}_{\text{pred}}$ is a single prediction point, $f(\boldsymbol{x}_{\text{pred}})$ is the predicted value calculated by the algorithm, b is the intercept or bias to the predicted function, and $\widetilde{\boldsymbol{w}}$ are the weights, given by

$$\widetilde{\boldsymbol{w}} = \boldsymbol{\alpha}^{\intercal} X, \tag{2.3}$$

with X as matrix containing the training points as row vectors like introduced in Subsection 2 on page 2, and $\boldsymbol{\alpha}$ as the optimised parameters. The matrix dimension is given by N. This linear expression can be extended to non-linear data by applying a non-linear mapping function $\Phi(\boldsymbol{x})$ that maps the input data into a high-dimensional feature space. In this new space the input data is again linear and the known algorithms can be applied [20]. In Equation 2.2 simply $\boldsymbol{x}$ is replaced with $\Phi(\boldsymbol{x})$, leading to the equation

$$f(\boldsymbol{x}) = \boldsymbol{w}\Phi(\boldsymbol{x}) + b. \tag{2.4}$$

The same transformation applies to the weights yielding,

$$\boldsymbol{w} = \boldsymbol{\alpha}^{\intercal}\Phi(X), \tag{2.5}$$

However, transforming every single data point into feature space and applying the fitting method is expensive and time-consuming. Fortunately, it is possible to avoid single-point transformations, as kernel based algorithms depend on dot products in feature space only which can be calculated with a so-called kernel $K(\boldsymbol{x_i}, \boldsymbol{x_j})$. This allows to evaluate the dot product in feature space between every input data point without transforming each point. In the kernel formulation, it has to be ensured that the kernel is valid and able to calculate the dot product in feature space. In principle

this means that the kernel has to be semi-positive definite and symmetric. This is equivalent to the constraint of satisfying Mercer's theorem [22],

$$\int \int f(\boldsymbol{x}_i) K(\boldsymbol{x}_i, \boldsymbol{x}_j) f(\boldsymbol{x}_j) \mathrm{d}\boldsymbol{x}_i \mathrm{d}\boldsymbol{x}_j \geq 0. \tag{2.6}$$

There are several different standard choices possible, which all satisfy Mercer's theorem. Most of them contain several hyper-parameters, which affect the training behaviour significantly. Hence, these parameters have to be selected carefully, for example via cross validation. Most popular are the radial basis function (RBF) or Gaussian [23], the polynomial (Poly), the linear (Lin) and the Matérn (Mat) [24][25] kernel,

$$K(\boldsymbol{x}_i, \boldsymbol{x}_j)_{\mathrm{RBF}} = \exp\{-\frac{\|\boldsymbol{x}_i - \boldsymbol{x}_j\|^2}{2\gamma}\}, \tag{2.7}$$

$$K(\boldsymbol{x}_i, \boldsymbol{x}_j)_{\mathrm{Poly}} = (\gamma \langle \boldsymbol{x}_i, \boldsymbol{x}_j \rangle + c)^d, \tag{2.8}$$

$$K(\boldsymbol{x}_i, \boldsymbol{x}_j)_{\mathrm{Lin}} = \langle \boldsymbol{x}_i, \boldsymbol{x}_j \rangle, \tag{2.9}$$

$$K(\boldsymbol{x}_i, \boldsymbol{x}_j)_{\mathrm{Mat}} = \frac{\sigma^2}{\Gamma(\nu) 2^{\nu-1}} (\gamma\sqrt{2\nu}) (\frac{\|\boldsymbol{x}_i - \boldsymbol{x}_j\|^2}{\gamma})^{\nu} K_{\nu}(\sqrt{2\nu}) \frac{\|\boldsymbol{x}_i - \boldsymbol{x}_j\|^2}{\gamma}, \tag{2.10}$$

with γ as a hyper parameter. The polynomial kernel also contains the parameter d, which defines the degree of polynomial, and c, a free parameter which defines the trade off between lower- and higher-order terms. The Matérn kernel contain the Gamma-function Γ and a modified Bessel-function K_ν. Both of them depend on the parameter ν, which has to be positive. In the limiting case of $\nu \to \infty$ the Matérn kernel converges to the Gaussian kernel.

It is also possible to train-kernel based methods simultaneously with function and gradient values. In order to do so, the kernel matrix is extended and the derivative of the kernel with respect to both input variables must exist [26][27],

$$\frac{\partial}{\partial \boldsymbol{x}_i} K(\boldsymbol{x}_i, \boldsymbol{x}_j) = K'(\boldsymbol{x}_i, \boldsymbol{x}_j), \tag{2.11}$$

$$\frac{\partial}{\partial \boldsymbol{x}_j} K(\boldsymbol{x}_i, \boldsymbol{x}_j) = K'(\boldsymbol{x}_i, \boldsymbol{x}_j)^{\intercal}, \tag{2.12}$$

$$\frac{\partial^2}{\partial \boldsymbol{x}_i \partial \boldsymbol{x}_j} K(\boldsymbol{x}_i, \boldsymbol{x}_j) = K''(\boldsymbol{x}_i, \boldsymbol{x}_j). \tag{2.13}$$

Using the matrix formalism introduced in Section Training Data on page 10 of this

thesis for the input data, the kernel matrix can be constructed as

$$A = \begin{bmatrix} K(X,X) & K'(X,X) \\ \\ K'(X,X)^\intercal & K''(X,X) \end{bmatrix}. \tag{2.14}$$

The weights have to be adapted via

$$\boldsymbol{w} = \begin{bmatrix} \Phi(X) \\ \Phi(X)' \end{bmatrix} \begin{bmatrix} \boldsymbol{\alpha} \\ \boldsymbol{\beta} \end{bmatrix}, \tag{2.15}$$

where $\boldsymbol{\beta}$ are derivative parameters that are optimised. With this extension it is possible to predict function and gradient values at a certain point. The prediction functions are given by

$$f(\boldsymbol{x}_{\text{pred}}) = \begin{bmatrix} K(X,\boldsymbol{x}_{\text{pred}}) \\ K'(X,\boldsymbol{x}_{\text{pred}}) \end{bmatrix} \begin{bmatrix} \boldsymbol{\alpha} \\ \boldsymbol{\beta} \end{bmatrix} + b, \tag{2.16}$$

$$f'(\boldsymbol{x}_{\text{pred}}) = \begin{bmatrix} K'(X,\boldsymbol{x}_{\text{pred}})^\intercal \\ K''(X,\boldsymbol{x}_{\text{pred}}) \end{bmatrix} \begin{bmatrix} \boldsymbol{\alpha} \\ \boldsymbol{\beta} \end{bmatrix}. \tag{2.17}$$

This gives the opportunity to give a prediction about information that is not included in the training process. For example if the algorithm is just trained to function values it is possible to predict gradient information. In this case the derivative weights $\boldsymbol{\beta}$ are zero and can be neglected.

2.2.1. Gaussian Process Regression (GPR)

This section follows roughly the book of Rasmussen et al. [3]. Gaussian processes are stochastic procedures that consist of a random variable collection. In the case of regression, these random variables are the different function values $y(\boldsymbol{x})$ at position $\boldsymbol{x}$. Each of them is described by a multivariate normal distribution. Any finite joint set of random variables is again a Gaussian distribution. The Gaussian process is completely defined by the mean and covariance function over the random variable. For simplicity, the function $\mathcal{N}(\overline{x}, \sigma^2)$ is introduced, which is a Gaussian function. It takes the mean $\overline{x}$ and variance σ^2 as arguments.

The procedure defines a Gaussian variable $f(x)$ with a mean and covariance function for each input. This covariance function measures the correlation between input data points. In general, Euclidean distances are used for the variable X. Since the

[3] Taken from [21], Chapter 2, page 7 et sequentes.

mentioned kernels measure the similarity between data points, the covariance function can be replaced by a kernel. This allows to extend the linear model to non-linear data. However, it should be noted that the fitting itself is still linear, but takes place in feature space. The two functions can be expressed as

$$\overline{x} = \mathbb{E}[f(X)], \tag{2.18}$$
$$\sigma^2 = K(X, X') = \mathbb{E}[(f(X) - \overline{x})(f(X') - \overline{x})]. \tag{2.19}$$

with $\mathbb{E}$ as the moment-generating function. Gaussian process regression uses Equation 2.4 in a Bayesian framework, where the weights are assumed to have a Gaussian distribution according to $\mathcal{N}(0, \Sigma_p)$. This function is also known as prior and can be expressed as

$$\mathbb{E}[f(X)] = \Phi(X)^\intercal \mathbb{E}[\boldsymbol{w}] \tag{2.20}$$
$$\mathbb{E}[f(X)f(X')] = \Phi(X)^\intercal \mathbb{E}[\boldsymbol{w}\boldsymbol{w}^\intercal]\Phi(X) = \Phi(X^\intercal)\Sigma_p\Phi(X') \tag{2.21}$$

where the covariance $\Phi(X^\intercal)\Sigma_p\Phi(X')$ can be identified as kernel $K(X, X')$. The mean is calculated by averaging over all input function values $\boldsymbol{y}(X)$. This value is also used for the interception term in Equation 2.4. Here the observed data points are assumed to be noise-free, but in general, they have noise with zero mean and σ_n^2 variance. This changes the covariance matrix slightly to

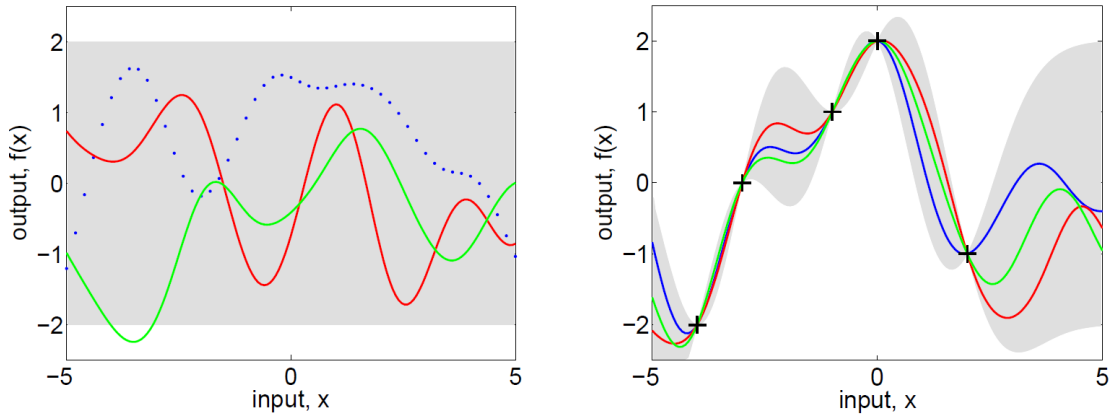
$$\sigma^2 = K(X, X') + I\sigma_\mathrm{n}^2, \tag{2.22}$$

where I is the identity matrix with ones in the main diagonal. From now on, every equation considers some noise which is added to the function value.

The joint prior distribution defines a confidence region where functions can be drawn from. Without knowledge of observations this will lead to randomly distributed functions, as it can be seen in Figure 2.4. To overcome this issue, the joint prior distribution has to select functions which are conform with the observations. Drawing random functions from the prior and rejecting those which are not in agreement is inefficient and expensive, but in probabilistic terms this can be easily done with the posterior function, which is given by

$$\begin{aligned} &p(f_\mathrm{pred}|\boldsymbol{x}_\mathrm{pred}, X, \boldsymbol{y}(X)) \sim \mathcal{N}\left(\overline{x}, \sigma_\mathrm{n}^2\right), \\ &\overline{x} = K(\boldsymbol{x}_\mathrm{pred}, X)[K(X, X) + I\sigma_\mathrm{n}^2]^{-1}\boldsymbol{y}(X), \\ &\sigma_\mathrm{n}^2 = K(\boldsymbol{x}_\mathrm{pred}, \boldsymbol{x}_\mathrm{pred}) - K(\boldsymbol{x}_\mathrm{pred}, X)[K(X, X) + I\sigma_\mathrm{n}^2]^{-1}K(X, \boldsymbol{x}_\mathrm{pred}). \end{aligned} \tag{2.23}$$

In these equations $\boldsymbol{x}_{\text{pred}}$ denotes new prediction points and f_{pred} is the predicted function value. By drawing from this distribution it is possible to make good predictions and give an appropriate error. This can be seen in the right panel of Figure 2.4. The posterior mean defines the predicted function value and the error is calculated by the sum of the covariance corresponding to the given sample.



(a) (b)

Figure 2.4.: Gaussian process regression influenced by no and five observation points. (a) Illustrates two functions randomly drawn from the joint prior distribution without any observations drawn as solid lines. The dotted line represents the true function curve. (b) Shows three functions drawn from the posteriori, where the joint prior distribution is conditioned on the five given data points. Pictures taken from Rasmussen [21] page 15.

Another positive aspect of Gaussian process regression is that the kernel hyperparameters can be optimised with the log-marginal likelihood. This function needs to be maximised and is calculated from the integral of prior times likelihood. By taking the logarithm of the integral the function is given by

$$\log(p(\boldsymbol{y}|X)) = -\frac{1}{2}\boldsymbol{y}(X)^{\intercal}(K(X,X)+I\sigma_{\text{n}}^2)^{-1}\boldsymbol{y}(X) - \frac{1}{2}\log|K(X,X)+I\sigma_{\text{n}}^2| - \frac{n}{2}\log(2\pi). \tag{2.24}$$

A major issue of this procedure is that it keeps all input values and does not dispose redundant data points, which may lead to expensive predictions.

It is also possible to extend the procedure to function and derivative value regression [3][2]. The kernel matrix has to be adopted similar to Equation 2.14. Since any noise in the derivative has to be different from noise in the function value, this leads to a

slightly different posterior function,

$$\begin{aligned} &p(f_{\text{pred}}|\boldsymbol{x}_{\text{pred}}, X, \boldsymbol{y}(X), \boldsymbol{y}'(X)) \sim \mathcal{N}\left(\overline{x}, \sigma_{\text{n}}^2\right), \\ &\overline{x} = \begin{bmatrix} K(\boldsymbol{x}_{\text{pred}}, X) \\ K'(\boldsymbol{x}_{\text{pred}}, X)^\intercal \end{bmatrix} K^* \begin{bmatrix} y(X) \\ y'(X) \end{bmatrix}, \\ &\sigma_{\text{n}}^2 = K''(\boldsymbol{x}_{\text{pred}}, \boldsymbol{x}_{\text{pred}})(K'(\boldsymbol{x}_{\text{pred}}, X)K^*K'(X, \boldsymbol{x}_{\text{pred}})), \end{aligned} \tag{2.25}$$

where the matrix K^* is given as

$$K^* = \begin{bmatrix} K(X,X) + I\sigma_{\text{n}}^2 & K'(X,X) \\ K'(X,X)^\intercal & K''(X,X) + I\sigma_{\text{d}}^2 \end{bmatrix}^{-1}, \tag{2.26}$$

with σ_{d}^2 as noise in the derivative.

2.2.2. Support Vector Regression (SVR)

The support vector machine formalism was first introduced by Vapnik et al. [4]. A detailed review on support vector regression is given by Smola et al. [20], which provided the main input for this section.

The key idea of a support vector machine is to represent the underlying function with support vectors from the input data. This is achieved by throwing away data points which contain redundant information. The basic formulation of a support vector machine is a simple linear fit and similar to Equation 2.2, but since it is a kernel based method, it is possible to extend the formalism by the introduction of kernels, which leads to Equation 2.4 for prediction.

Support vector regression can be used in two different ways: One is to keep a certain number of support vectors, which introduces an unknown error. The other is to keep the fitted curve close to a specific error value on each input data point. This technique is called ϵ-support vector regression. This formulation does not allow to control the number of support vectors in the end, but it only tolerates an error ϵ around the fitted curve. It looks like two parallel curves with an offset of ϵ above and beneath the actual predicted curve. The error is then given by

$$\begin{aligned} \boldsymbol{y}(X) - \Phi(X)^\intercal \boldsymbol{w} - b - \epsilon &\leq \xi, \\ -\boldsymbol{y}(X) + \Phi(X)^\intercal \boldsymbol{w} + b - \epsilon &\leq \xi^*, \\ \xi,\xi^* &\geq 0, \end{aligned} \tag{2.27}$$

where the lower boundary is indicated by the superscript *, ϵ is the insensitive region around the fitted curve, and ξ is the upper and ξ^* is the lower slack variable which define the actual error on each side. The relation between these two parameters can be seen in Figure 2.5.

Since two boundaries exist also the weight consists of two parameters α and α^* which are restricted to a positive value. This leads to a modification of Equation 2.5, yielding

$$\boldsymbol{w} = \Phi(X)^\intercal(\boldsymbol{\alpha} - \boldsymbol{\alpha}^*), \tag{2.28}$$

where α is the upper and α^* is the lower coefficient of support vectors. A well fitted curve is obtained by minimising the error and squared weight norm, since minimising only the error will lead to overfitting. This leads directly to the primal optimisation

[4] An introduction is given in [28] Chapter 6, page 181 et sequentes.

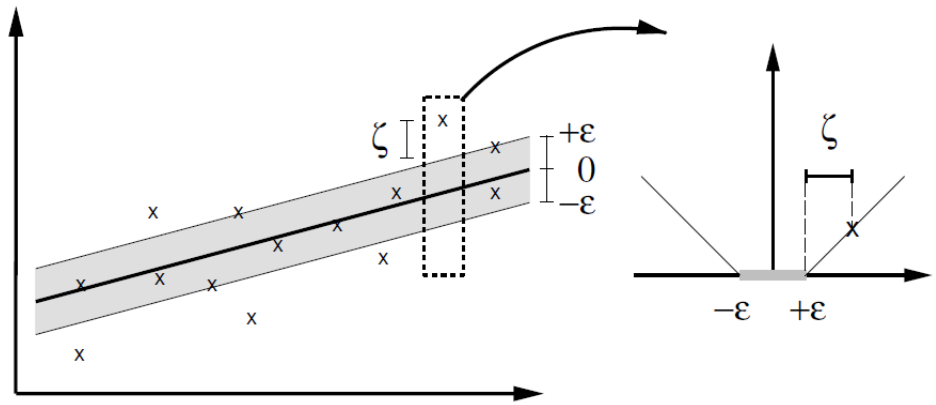


Figure 2.5.: This figure shows the relation between the insensitive region ϵ around the fitted curve and the slack variables $\xi^{(*)}$ [20].

problem of the support vector formulation, the minimisation of the Lagrangian function, which can be expressed as

$$\mathcal{L} = \frac{1}{2}\boldsymbol{w}^\intercal \boldsymbol{w} + C\sum_{i=1}^{N}(\xi_i + \xi_i^*), \tag{2.29}$$

which is subject to Equation 2.27. The parameter C represents the regularisation constant and defines which error has more impact on the minimisation procedure. In most cases, the primal problem is easier to solve in the dual formulation, where all constrains are included into the optimisation function. Solution to the dual problem is obtained by applying the Krush-Kuhn-Tucker (KKT) condition [29], which states that the optimal solution in the dual space is equivalent to the solution in primal space. The result of this procedure is given by

$$\text{maximise} \left\{ -\frac{1}{2}\boldsymbol{w}^\intercal \boldsymbol{w} - \sum_{i=1}^{N}\epsilon(\alpha_i + \alpha_i^*) + \sum_{i=1}^{N} y_i(\alpha_i - \alpha_i^*) \right. \quad , \tag{2.30}$$

$$\text{subject to} \left\{ \begin{array}{l} \sum_{i=1}^{N}(\alpha_i - \alpha_i^*) = 0 \\ \alpha_i, \alpha_i^* \in [0, C] \end{array} \right. . \tag{2.31}$$

Intermediate steps from the primal optimisation problem to the dual formalism can be found in Appendix A.5. The dual problem solution is outlined in Appendix A.6.

It is also possible to include derivative information in the formalism by introducing a set of error variables for the derivatives [26]. Again, these split into upper and lower

boundaries,

$$\begin{aligned} \boldsymbol{y}(X)' - \Phi(X)'\boldsymbol{w} - \epsilon' &\leq \tau, \\ -\boldsymbol{y}(X)' + \Phi(X)'\boldsymbol{w} - \epsilon' &\leq \tau^*, \\ \tau^{(*)} &\geq 0, \end{aligned} \tag{2.32}$$

where $\tau(*)$ are the slack variables for the derivatives and ϵ' is the insensitive width region for derivatives. Since derivative values are included, the weight function looks similar to Equation 2.15, but with the small difference that $\boldsymbol{\alpha}$ consists of $\boldsymbol{\alpha} - \boldsymbol{\alpha}^*$. Similar rules apply to the $\boldsymbol{\beta}$ of Equation 2.15. With this modification, the primal optimisation problem can be expressed as

$$\text{minimimze} \quad \left(\frac{1}{2}\boldsymbol{w}^\intercal \boldsymbol{w} + C\sum_{i=1}^{N}(\xi_i + \xi_i^*) + D\sum_{i=1}^{N}(\tau_i + \tau_i^*) \right), \tag{2.33}$$

where the optimisation is subject to Equation 2.27 and 2.32.

Transformation steps from primal to dual problem are explained in the Appendix A.5. In Appendix A.6 is described how to obtain the solution to the dual problem. Here only the solution to the dual problem shall be given,

$$\text{maximise} \begin{cases} -\frac{1}{2}\boldsymbol{w}^\intercal \boldsymbol{w} - \sum_{i=1}^{N}\epsilon(\alpha_i + \alpha_i^*) + \sum_{i=1}^{N} y_i(\alpha_i - \alpha_i^*) \\ -\sum_{i=1}^{m*N}\epsilon'(\beta_i - \beta_i^*) + \sum_{i=1}^{m*N} y_i'(\beta_i - \beta_i^*) \end{cases}, \tag{2.34}$$

$$\text{subject to} \begin{cases} \sum_{i=1}^{N}(\alpha_i - \alpha_i^*) = 0 \\ \alpha_i, \alpha_i^* \in [0, C] \qquad \beta_i, \beta_i^* \in [0, D] \end{cases}, \tag{2.35}$$

where D is the regularisation parameter for derivatives.

Since the dual problem solution is a quadratic programming problem, there are several methods to solve it. This thesis focuses on iterative re-weighted least squares and regularised least squares formulation. From now on every algorithm will consider function and derivative values.

Regularised Least Square (RLS)

The regularised least squares procedure redefines the quadratic programming problem in such a way that only a set of linear equations has to be solved. This was introduced by Suykens et al. [30]. Jayadeva et al. [27] extended this approach to derivative values, which is the main concern of this section.

The regularised least squares is a rather simple approach. It reformulates the primal problem in such a way that a set of linear equations is obtained. The trick is to employ equality constrains to Equation 2.33 and consider only one side of the boundary. This is allowed, because the error can also be negative in this formulation. Therefore, only one error variable for each value is considered, which leads to a simple primal problem given by

$$\text{minimise } \left(\frac{1}{2}(\boldsymbol{w}^\intercal \boldsymbol{w}) + \frac{C}{2}\boldsymbol{\xi}^\intercal\boldsymbol{\xi} + \frac{D}{2}\boldsymbol{\tau}^\intercal\boldsymbol{\tau}\right) \tag{2.36}$$

$$\text{subject to } \begin{cases} \Phi(X)^\intercal \boldsymbol{w} + b - \boldsymbol{y}(X) + \boldsymbol{\xi} = 0 \\ \Phi(X)'^\intercal \boldsymbol{w} - \boldsymbol{y}(X)' + \boldsymbol{\tau} = 0 \end{cases} \tag{2.37}$$

with $\boldsymbol{\xi}$ and $\boldsymbol{\tau}$ as error variables. Thereby it is possible to formulate the dual Lagrangian, see Appendix A.5. By using the Karush-Kuhn-Tucker condition, see Appendix A.6, it is possible to eliminate the primal variables, which leads to the following set of equations,

$$\begin{bmatrix} K(X,X) + I\frac{1}{C} & K'(X,X) & \mathbf{1} \\ K'(X,X)^\intercal & K''(X,X) + I\frac{1}{D} & \mathbf{0} \\ \mathbf{1} & \mathbf{0} & 0 \end{bmatrix} \begin{bmatrix} \boldsymbol{\alpha} \\ \boldsymbol{\beta} \\ b \end{bmatrix} = \begin{bmatrix} \boldsymbol{y}(X) \\ \boldsymbol{y}(X)' \\ 0 \end{bmatrix}, \tag{2.38}$$

where I is an identity matrix with appropriate dimensions, $\mathbf{1}$ is a vector containing only ones with dimension N, and $\mathbf{0}$ is the zero vector with dimension N.

This set of linear equations can be easily solved by inverting the matrix and multiplying the whole equation from the left with it. Regularised least squares uses every input data point as a support vector due to simplification at the beginning. Thus, it loses the main advantage of support vector machine algorithms and is more related to the ridged regression method. In addition to that, the regularised least squares approach has no error-insensitive region. However, for moderate system sizes it is supposed to give good results in a reasonable time.

Iterative Re-weighted Least Squares (IRWLS)

The section follows roughly the proposed procedure by Lazáro et al. [26]. The procedure is iterative and converges to the real support vector machine solution. It works well for moderate sized problems. On larger systems it will get computationally intensive and therefore very expensive, because the procedure has to invert a matrix in every iteration.

The iterative re-weighted least squares approach uses Equation 2.33, where the given constraints are restricted to be equal and the slack variable constraints are replaced with the maximum function $\max(u, 0)$ with $u = \xi, \xi^*, \tau, \tau^*$. The algorithm uses a second order Taylor expansion over the slack variables. Intermediate steps to this procedure are outlined in Appendix A.7. The maximum function has to be replaced with a derivable approximation, given by

$$\max(u, 0) \approx L(u) = \begin{cases} 0, & u < 0, \\ \frac{K}{2}u^2, & 0 \leq u \leq \frac{1}{K} \\ u - \frac{1}{2K}, & u \geq \frac{1}{K} \end{cases}. \tag{2.39}$$

For $K \to \infty$ this equation tends to the maximum function. With this, the second order Taylor expansion, also called quadratic approximation, can be written as,

$$\frac{1}{2}\boldsymbol{w}^\intercal \boldsymbol{w} + \frac{1}{2}\sum_{i=1}^{N} a_i \xi_i^2 + a_i^* \xi_i^{*2} + s_i^* \tau_i^2 + s_i^* \tau_i^{*2} + \text{const.}, \tag{2.40}$$

where the constant is independent of the weights $\boldsymbol{w}$ and the interception b. Therefore, it can be neglected. The variables a_i are the upper and a_i^* are the lower weights to the function value error and equivalent to that the variables s_i and s_i^* are defined as the derivative error weights. These error weights follow the same definition and can be calculated via

$$d_i^{(*)} = \begin{cases} 0, & e_i^k < 0 \\ KC_{\text{reg}} & 0 \leq e_i^k \leq \frac{1}{K} \\ \frac{C_{\text{reg}}}{e_i^k} & e_i^k \geq \frac{1}{K} \end{cases} \tag{2.41}$$

where d_i is the error weight, C_{reg} is the regularisation constant to the given error, e_i^k is the calculated error in the k-th step and the subscript i defines the sample. For an

arbitrarily large K the term in the middle vanishes.

Form the error definition in Equation 2.33, with equal constrains, one can see that the upper and lower boundary error can not be positive at the same time. Also both corresponding error weights can not be positive at the same time, but the two of them can be zero. If both of the two related error weights are zero, $a_i = a_i^* = 0$ or $s_i = s_i^* = 0$, the sample i is not a support vector and therefore ignored in Equation 2.40.

Minimising Equation 2.40 is done by using the Karush-Kuhn-Tucker condition as outlined in Appendix A.6. This leads to the following solution, written in matrix form,

$$\begin{bmatrix} K(X,X) + I\frac{1}{\boldsymbol{a}+\boldsymbol{a}^*} & K'(X,X) & \mathbf{1} \\ K'(X,X)^\intercal & K''(X,X) + I\frac{1}{\boldsymbol{s}+\boldsymbol{s}^*} & \mathbf{0} \\ \mathbf{1} & \mathbf{0} & 0 \end{bmatrix} \begin{bmatrix} \boldsymbol{\alpha} - \boldsymbol{\alpha}^* \\ \boldsymbol{\beta} - \boldsymbol{\beta}^* \\ b \end{bmatrix} = \begin{bmatrix} \boldsymbol{y}(X) + \frac{\boldsymbol{a}-\boldsymbol{a}^*}{\boldsymbol{a}+\boldsymbol{a}^*} \\ \boldsymbol{y}(X)' + \frac{\boldsymbol{s}-\boldsymbol{s}^*}{\boldsymbol{s}+\boldsymbol{s}^*} \\ 0 \end{bmatrix}, \quad (2.42)$$

where I is an identity matrix with appropriate dimensions. It is possible to collect the two different variables $\boldsymbol{\alpha}$ and $\boldsymbol{\alpha}^*$ into a single variable $\boldsymbol{\alpha}$. The same applies to $\boldsymbol{\beta}$ and $\boldsymbol{\beta}^*$. Note that only those samples have to be considered that have a non-zero error weight. Due to the insensitive regions ϵ and ϵ', this reduces the size of the matrix if redundant information is present.

The whole algorithm can be summarised as follows:

1. Initialise the error weights.
2. Solve Equation 2.42 for $\boldsymbol{\alpha}$, $\boldsymbol{\beta}$ and b by inverting the matrix.
3. Evaluate the errors and recalculate the error weights.
4. Calculate the Lagrangian of Equation 2.40 and check if the difference to the previous step is below a certain threshold. If not, continue with step 2.

3. Results

In chemistry the knowledge of reactions is essential. Experimentally, it is exhausting and expensive to determine reaction rates and selectivities of chemical processes under varying conditions. Here, computer experiments, can provide valuable information by an efficient exploration of the underlying potential energy surface. On this landscape the reaction is described by the minimum energy path which connects two minima, the reactant and product. The highest point on this pathway is the transition state which defines the reaction rate and separates the two minima. Several methods are well-established for the transition state search, but they all rely entirely on *ab initio* calculations. Thus, such a search is time-consuming, especially in case of larger systems. Obviously, these algorithms would operate much faster if the number of *ab initio* calls could be reduced somehow.

3.1. Preliminary Work

A necessary pre-requisite for the intended acceleration of the transition state search is a method which is capable of locating the transition state. For this the nudged elastic band method is implemented, since it allows to approximate the minimum energy path and is able to locate the transition state as well. Additionally, to test implemented algorithms, a two-dimensional test surface is introduced. In the following, this landscape will be described first. After that, the nudged elastic band technique is discussed.

3.1.1. 2D-Test Surface

The two-dimensional test surface represents a fictitious potential energy landscape defined by an analytic function, which allows for a quick evaluation of the energy at any point. In the literature, the Müller-Brown potential energy surface is typically used for that purpose [31]. The function for this landscape is given by

$$E(x,y) = \sum_{i=0}^{4} A_i \, e^{a_i(x-\bar{x}_i)^2 + b_i(x-\bar{x}_i)(y-\bar{y}_i) + c_i(y-\bar{y}_i)^2}, \tag{3.1}$$

where x and y are the coordinates and E is the energy value at specific coordinates. All other parameters $(A_i, a_i, b_i, c_i, \bar{x}_i, \bar{y}_i)$ are kept fixed and are listed in Table 3.1.

The Müller-Brown surface consists of three minima and two transition states. The coordinates of these special points are tabulated in Table 3.2. It should be noted that

Table 3.1.: Parameters for the Müller-Brown potential energy surface.

i	A	a	b	c	$\bar{x}$	$\bar{y}$
1	−200	−1.0	0	−10	1	0
2	−100	−1.0	0	−10	0	0.5
3	−170	−6.5	11	−6.5	−0.5	1.5
4	15	0.7	0.6	0.7	−1	1

the Müller-Brown surface consists only of two minimum energy paths, because the third pathway is equivalent to the connected path of the other two by construction.

Table 3.2.: Analytically calculated coordinates of the three minima and two transition states on the Müller-Brown potential energy surface.

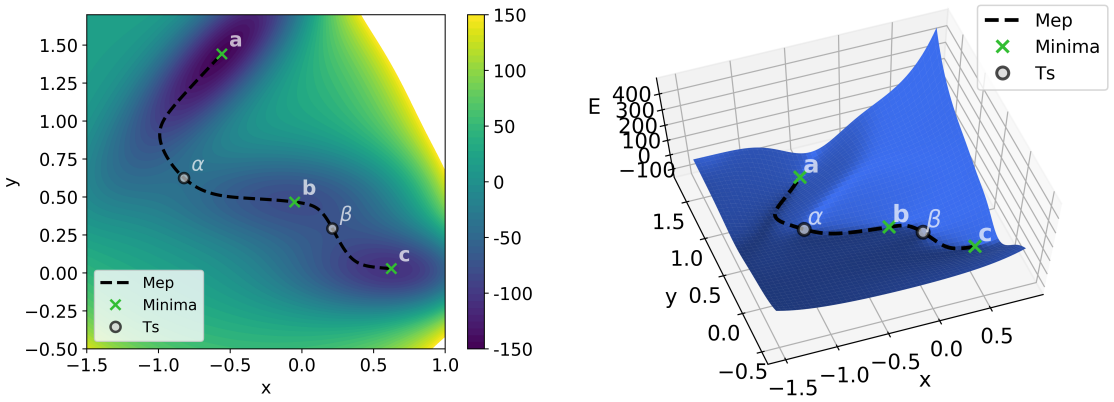
Name	x	y
Minimum a	−0.558 224	1.441 730
Minimum b	−0.050 010	0.466 694
Minimum c	0.623 499	0.028 037
Transition state α	−0.822 002	0.624 313
Transition state β	0.212 487	0.292 988

Since the landscape is analytically derivable it allows to calculate both minimum energy paths exactly. In Figure 3.1 the Müller-Brown surface is illustrated with the minima, transition states and the minimum energy pathways. There it can be seen that the first minimum energy path connects minimum a and b via the transition state α. The second path starts at minimum b and ends at minimum c. In between lies the transition state β. Both pathways touch each other in minimum b, so that the minimum energy path from a to c is equivalent to the two connected pathways.

3.1.2. Nudged Elastic Band Tests

The transition state search is done via the nudged elastic band method, which is capable of approximating the minimum energy path as well as the transition state. Note that the algorithm first converges to the minimum energy path and then it locates the transition state on that pathway. Different states on the potential energy surface are referred to as images.

At the beginning of the nudged elastic band algorithm both initial minima are connected via a linear interpolation in coordinate space, which consists of N equidistant images along the path. Increasing the number of intermediate images will improve the minimum energy path approximation, but also increases the number of energy and gradient



(a) 2D perspective contour

(b) 3D perspective

Figure 3.1.: Müller-Brown potential energy surface with the minimum energy path that connects all three minima (a, b, c) and the two transition states α, β. The coordinates of them are given in Table 3.2.

evaluations per iteration. Typically an uneven number is used, as the transition state can be assumed to be located in the middle of the minimum energy pathway in general.

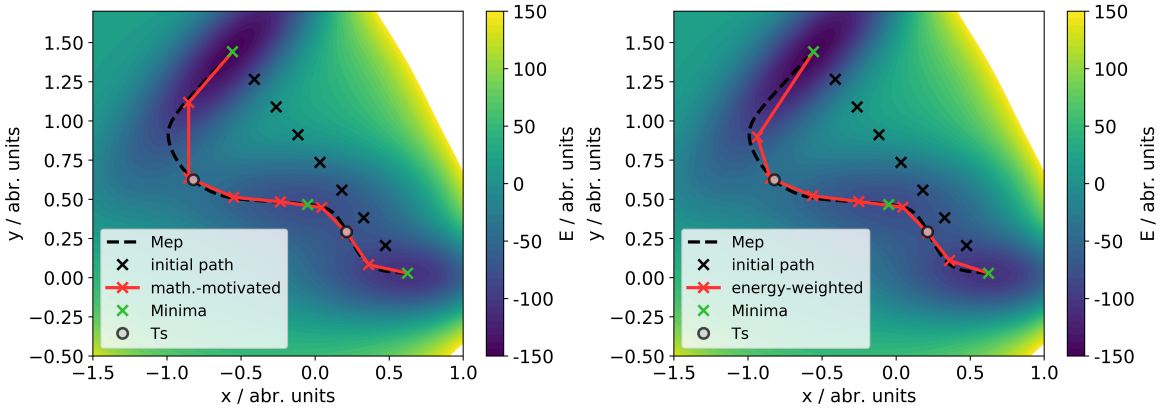
Another important parameter is the optimisation method which affects the speed of convergence and the stability of the algorithm significantly. Even if the nudged elastic band is a standard minimisation problem, it is not possible to use every optimisation method, because gradient evaluations on the potential energy surface are computationally expensive and shall therefore be avoided. The simplest optimisation algorithm is the *steepest decent* [1], but it is known that this algorithm has a slow convergence and tends to oscillations near the minimum. Thus, two other optimisation methods have been implemented, namely the *fast inertial relaxation engine* (Fire) [33] and *Quick-Min* [34] using the *velocity verlet algorithm* [35]. Preliminary tests, which are not explicitly presented here, have shown that both methods are comparable with respect to stability and convergence speed. Thus, for all further calculations the *fast inertial relaxation engine* is used.

Additionally, the nudged elastic band convergence and stability is also affected by the tangent estimation method, which in turn might change the obtained minimum energy path approximation. Figure 3.2 shows the minimum energy path obtained by the improved mathematically-motivated and the energy-weighted tangent estimation. Both calculations are done with the same set of parameters, which is listed in Table 3.3. The mathematical derivations of the tangent estimations are outlined in Appendix A.3. As

[1] For a description of steepest descent see Jensen [32], page 383.

it can be seen in Figure 3.2b, the energy-weighted method neglects some of the positive effects of projecting the spring force into parallel and the gradient perpendicular to the band. Thus, it allows non-equidistant images. This effect can be seen at the upper minimum a, where the band is stretched. Additionally, it allows the nudged elastic band method to underestimate sharp curves of the minimum energy path.

The mathematically-motivated approach, on the other hand, keeps the images at equal distances. However, it should be mentioned that this tangent estimation approach tends to a kinky path in cases where the parallel component of the gradient is large compared to the perpendicular component and, furthermore, is does not converge to the minimum energy path but starts to oscillate [11].



(a) Mathematically-motivated tangent estimation

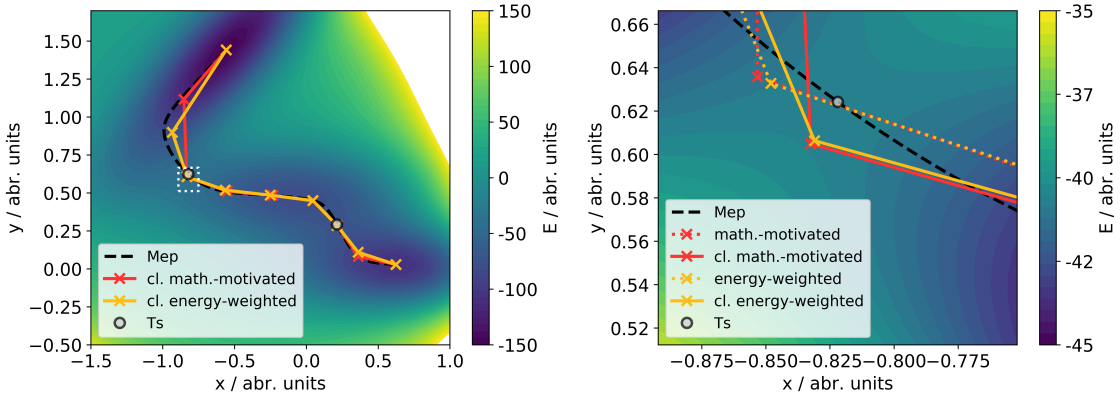
(b) Energy-weighted tangent estimation

Figure 3.2.: Approximation of the minimum energy path with two different tangent estimation methods. (a) The mathematically-motivated estimation keeps the distances between the images better at equal distances. (b) The energy-weighted tangent estimation neglects some positive effects of the gradient and spring force projection. Thus, it allows to stretch the band. Additionally, corner-cutting becomes possible, see the upper minimum and the neighbouring image.

Table 3.3.: Nudged elastic band parameters on the Müller-Brown surface for the mathematical-motivated and energy-weighted tangent estimation method using the *fast inertia relaxation engine*.

Search	ΔT	ΔT_{max}	r_{trust}	spring	conv. grad.
Mep	3.5	7.0	0.001	0.001	0.01
Ts	3.5	7.0	0.001	0.001	0.005

After approximating the minimum energy path, the nudged elastic band method is able to locate the highest transition state on that pathway. In order to do so, the image with the highest energy has to climb uphill by following the positive gradient. This will directly lead the image to a maximum, hence, the image follows the parallel gradient component. Furthermore, the spring force for this image is set to zero, so that gradient and spring force do not cancel each other and the image can move freely.
Both tangent estimations are able to locate the transition state, as can be seen in Figure 3.3. Figure 3.3a shows the complete minimum energy path, while Figure 3.3b zooms into the vicinity of the transition state. Additionally, the graphs further contain the minimum energy path approximation before application of the climbing image to illustrate differences. In it can be seen that both tangent estimation methods are capable of locating the transition state, even if the minimum energy path approximation differs slightly. All nudged elastic band parameters for the calculations are listed in Table 3.3.



(a) Overview over the whole climbing minimum energy path.

(b) Vicinity of the transition state.

Figure 3.3.: Locating the transition state with the nudged elastic band method for the two different tangent estimation methods. The left graphics (a) shows the complete minimum energy path. The right graphic (b) illustrates the vicinity of the transition state. Additionally, it shows the path from that the climbing algorithm starts. As aspected, the climbing image climbs uphill.

3.2. Machine Learning

Machine learning techniques are able to represent high dimensional scalar functions. Thus, these algorithms can be used to mimic the molecular potential energy surface and to accelerate transition state searches. With this, the question arises which information can be provided without further *ab initio* calculations and which data should be used in the training process. Transition state search via the nudged elastic band method uses energy and gradient information of the molecular potential energy surface. Thus, these data can be used in the training process of the machine learning technique, but using energy and gradient values will also increase the training effort.

That does not answer the question which information should be used to train the algorithm. Normally, providing more data points should lead to a better prediction result. Thus, it has to be proved that including gradient information in the training process is worth the effort. For that purpose, a one-dimensional scalar function is used and the regularised least squares algorithm is trained exclusively, since all other implemented algorithms will show a similar behaviour The training parameters are summarised in Table 3.4. The one-dimensional scalar function is given by

$$f(x) = 3\sin(x)\, e^{-\frac{x^2}{225}} + 2. \tag{3.2}$$

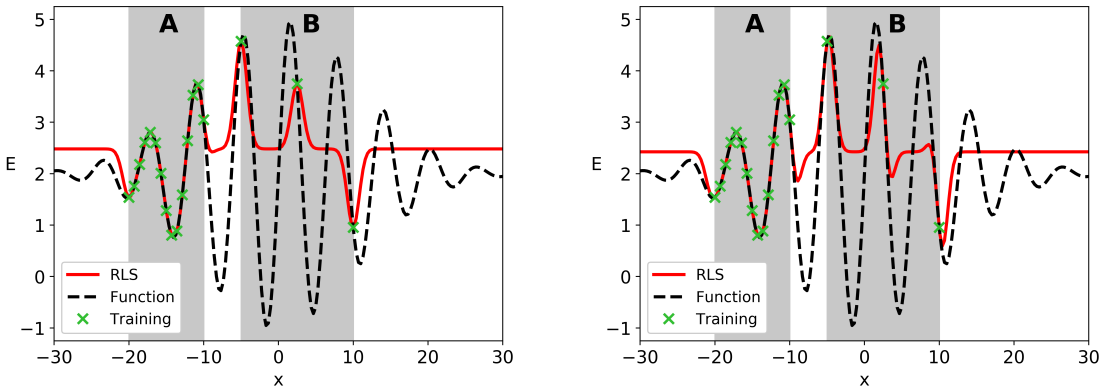
Table 3.4.: Regularised least squares parameters for the training on the one-dimensional function.

available information	kernel	σ	value reg.	grad. reg.
value	RBF	0.8	10^3	-
value and gradient	RBF	0.8	10^3	10^3

The training set is split into two regions with different training point density, but in both regions the points are linearly distributed. Region A is dense, starts at -20, ends at -10 and contains 15 training points. The second region B has only 3 training points, starts at -5 and ends at 10. In Figure 3.4 these two regions A and B are shaded in grey.

First, the regularised least squares method is trained only with energy values, with the result shown in Figure 3.4a. It can be seen that in region A the underlying function and its derivative are well represented by the machine learning technique. When looking at region B, it can be seen that the training point values are well reproduced by the machine learning prediction, but the actual functional behaviour is not captured at all. Outside of both regions the machine learning algorithm predicts a constant value.

Figure 3.4b shows the machine learning prediction after training with energy and gradient information. There it can be seen that the machine learning technique represents the function curve well in region A, similar to the training with only energy information. In region B it reproduces the function values at the training points reasonably, but now also the correct functional behaviour in the vicinity of these points is provided by the machine learning algorithm.



(a) Value information

(b) Value and gradient information

Figure 3.4.: Comparison between the available information in the regularised least squares training process. In both the same training points are used, which are linearly distributed in two regions. Region A starts at -20 and ends at -10 with 15 training points. The second region B consists of 5 training points and starts at -5 and ends at +10. (a) shows the training with only value information. (b) illustrates training with value and gradient information.

By comparing the two training techniques it is obvious that both show a similar behaviour in region A and on the outside of both regions. Furthermore, it has shown that including gradient information allows to reproduce curvatures more accurate than by training the algorithm just with energy values. Outside of both regions, where no information is present in the training set, both training techniques predict a constant value. From this it can be concluded that machine learning techniques interpolate well, but are not able to extrapolate. Furthermore, the machine learning method needs less training points to represent the underlying function, if gradient information is included in the training process.

Additionally, it should be mentioned that training with only gradient information would be also possible in principle, but gives rise to two major issues: First, the predicted curvature of the machine learning algorithm will not be able to represent the function value due to the offset. Second, amplitudes will be dramatically wrong, since gradient

information defines only the curvature slope. Thus, this training method is neglected and not shown explicitly. In the following, all training processes use both energy and gradient information.

3.2.1. Reaction Path Search Via Machine Learning

Although the training of the machine learning technique implies some computational overhead, it is possible to accelerate the reaction path search by reducing the number of *ab initio* calculations needed. This is possible because predictions based on the machine learning technique are cheap compared to *ab initio* evaluations.

For that, the reaction path search is done on a fictitious potential energy surface provided by the machine learning technique. The transition state search process is shown schematically in Figure 3.5. After convergence on the predicted fictitious surface, new geometries are obtained. However, in order to guarantee that the reaction path of the molecular potential energy surface is correctly approximated, gradient and energy of each newly obtained geometry are calculated on the real molecular potential energy surface. If the convergence criteria are also met on the latter the real reaction path is located. If not, these new geometries are added to the training set in order to improve at least the next prediction. However, for the reaction path search via machine learning the following steps are necessary:

1. Initialise a linear interpolated path between the two minima.
2. Calculate energy and gradient of each image on this initial path.
3. Store the geometries together with calculated energy and gradient values.
4. Train the machine learning algorithm with all stored values.
5. Start from the initial path and apply the nudged elastic band method by using the energy and gradient of the assumed potential energy surface of the machine learning technique.
6. Calculate energy and gradient on the *ab initio* potential energy surface for each new obtained geometry.
7. Check if all gradients are below a given threshold, otherwise go to step 3 add the new values and gradients to the training set, and continue with the algorithm until convergence.

A similar procedure applies to locating the transition state, which can be done after the minimum energy path has been approximated. For that the spring force of the geometry with the highest energy is set to zero to make it move freely. It now follows the parallel gradient component with respect to the band instead of the perpendicular

component. Additionally, the initial pathway is set to the minimum energy path. For a detailed example calculation on the Müller-Brown surface the reader is referred to Appendix A.8, where predicted fictitious potential energy landscapes during the reaction path search process are shown.

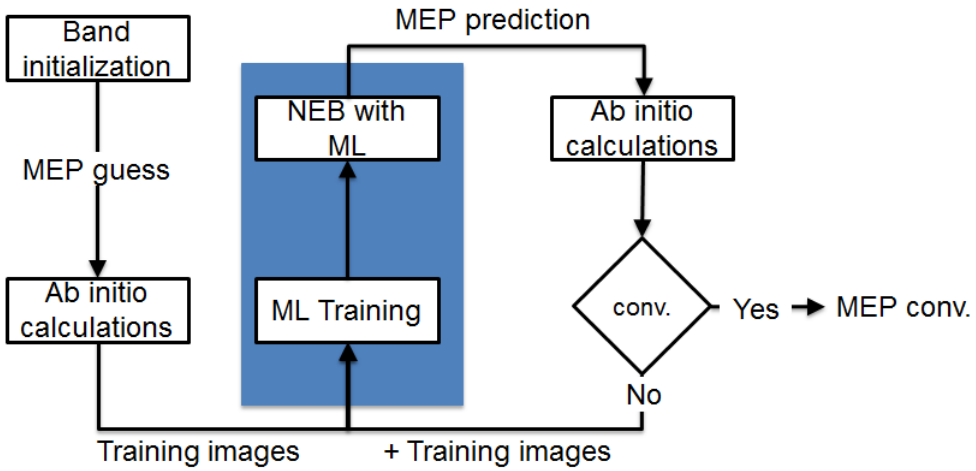


Figure 3.5.: Scheme of the reaction path search process via machine learning techniques.

3.2.2. Müller-Brown Test Surface

The machine learning techniques are tested on the two-dimensional Müller-Brown surface. Note that the neural network approach is excluded from this study, because it uses descriptor functions as proposed by Behler et al. [19], which take molecular symmetries into account. Hence, the neural network approach can only be applied to three-dimensional structures and can not be applied to the Müller-Brown surface.

All calculations in this subsection are done with the same nudged elastic band parameter set which is listed in Table 3.5. All machine learning parameters used are tabulated in Table 3.6. Additionally, it should also be noted that the same convergence threshold is used for the nudged elastic band method on the predicted fictitious landscape as on the Müller-Brown surface.

Table 3.5.: Nudged elastic band parameters used on the fictitious machine learning surface.

Search	tangent method	ΔT	ΔT_{max}	r_{trust}	spring	conv. grad
Mep	math. motivated	3.5	7.0	0.005	10^{-5}	0.01
Ts	math. motivated	3.5	7.0	0.005	10^{-5}	0.005

Table 3.6.: Machine learning parameters for the Müller-Brown surface.

method	kernel	σ	C_{value}	C_{grad}	restarts	norm. y	ϵ	ϵ'	eps
RLS	RBF	0.196	10^{10}	10^{10}	-	-	-	-	-
GPR	RBF	0.196	10^{10}	10^{10}	5	No	-		-
IRWLS	RBF	0.196	10^{7}	10^{7}	-	-	10^{-5}	10^{-5}	10^{-6}

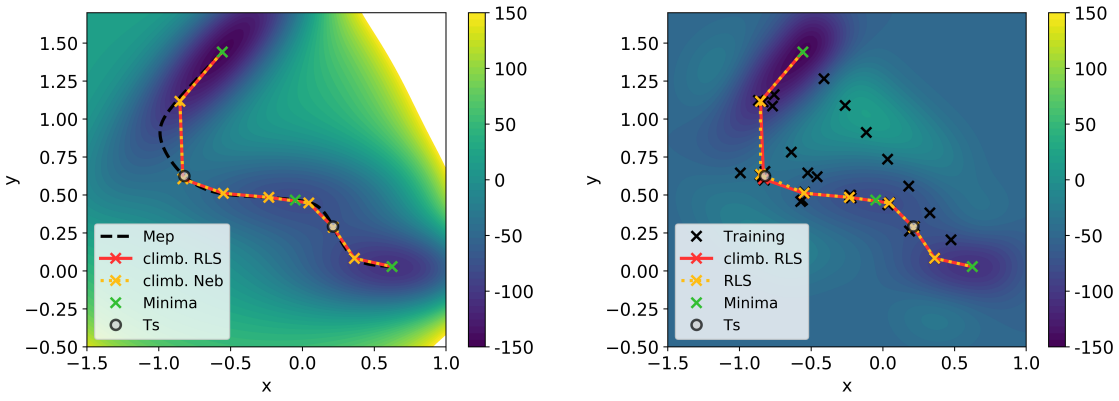
In the following, the result of each kernel method is discussed first. At the end of this section, all of them are compared with each other.

Regularised least squares

The regularised least squares method can be easily implemented. It uses all training points for the training and predictions. Therefore, the computational expenses rise for both with increasing training set size. Furthermore, in this method the kernel hyper-parameter is a constant value and does not change during the transition state search process. Optimisation of this parameter is expensive and can be done for example via a grid search. However, a non-optimal hyper-parameter does not affect the algorithm stability, but slows down the convergence speed of the transition state search, since qualitatively representation of the potential energy surface suffers. The algorithm stability is mostly influenced by the two regularisation parameters. In order to represent the training data well these two parameters have to be larger, but at about a value of 10^{12} the algorithm gets unstable due to the matrix inversion. All used machine learning parameters are listed in Table 3.6.

In Figure 3.6 the Müller-Brown surface is compared with the predicted landscape by the machine learning technique after it has located the transition state. It can be seen that the minimum energy path as well as the transition state are reasonably well located and a path similar to that obtained with the plain nudged elastic band method is obtained, see Figure 3.6a. The predicted fictitious surface is shown in Figure 3.6b. It also contains all training points calculated until the minimum energy path is obtained, which represents the training set necessary to obtain the minimum energy path. Further evaluations which are needed to obtain the transition state are skipped in the graph as they are too close together and therefore undistinguishable.

By comparing the Müller-Brown surface to the predicted fictitious landscape it is obvious that the vicinity of the minimum energy path is well represented by the machine learning algorithm. Outside, where no information about the real surface is available, the algorithm predicts a constant value. The regularised least squares method has approximated the minimum energy path within 5 iterations and located the transition state after 8 further iterations.



(a) Müller-Brown surface (b) Fitted surface

Figure 3.6.: Comparison between the Müller-Brown surface and the predicted fictitious landscape by regularised least squares technique. (a) Comparing the reaction path of the plain nudged elastic band with the obtained path with the regularised least squares approach. (b) Illustrates the predicted surface after climbing. There are only these training points visible which are needed to obtain the minimum energy path.

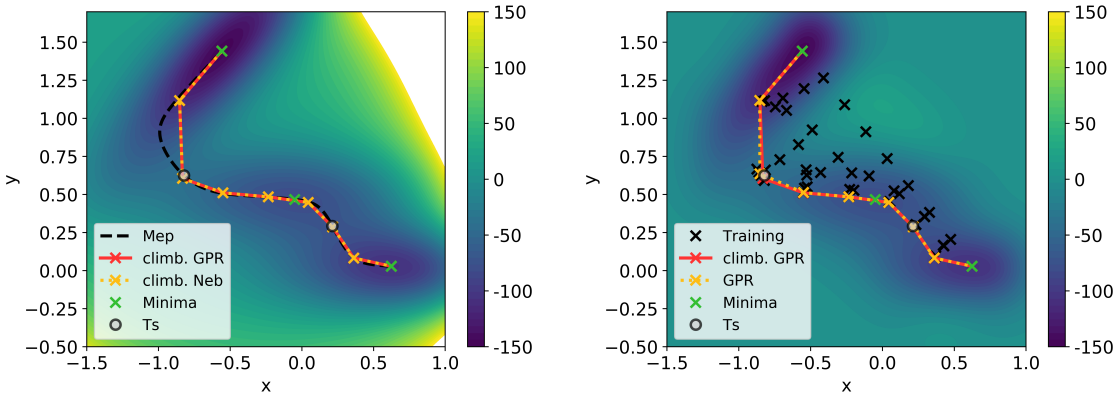
Gaussian process regression

The implementation of Gaussian process regression takes more effort than the regularised least squares approach. However, Gaussian process regression implements a procedure to optimise the kernel hyper-parameter. Therefore it uses the derivative of the kernel with respect to the hyper-parameter. For the optimisation the first training set is used always, because it has shown that the optimisation with another training set leads to a similar kernel hyper-parameter. Hence, to reduce computation time the hyper-parameter is optimised only once with the smallest possible training set. The optimisation restarts five times with different hyper-parameter values to obtain the global minimum. Initial hyper-parameter values are randomly drawn from the range -11.513 to 11.513. Table 3.6 lists the Gaussian process regression parameters with the optimised kernel hyper-parameter. It should be noted that the Gaussian process regression formalism does not employ regularisation, but contains noise (see Section 2.2.1). Since these two parameters are inversely related to each other, only the regularisation is listed in Table 3.6.

Figure 3.7 shows the potential energy landscape comparison between the Müller-Brown surface and the predicted fictitious landscape. It can be seen in Figure 3.7a that the nudged elastic band method based on Gaussian process regression obtains a similar minimum energy path as the plain nudged elastic band method does. Both methods identify the same transition state. The predicted fictitious surface after locating the

transition state is shown in Figure 3.7b, where all training points to obtain the reaction path are again included. As before, further training points used to locate the transition state are not shown, as they lie very close together

A comparison of both potential energy surfaces shows that the Müller-Brown surface is well represented close to the minimum energy path by the machine learning prediction. Where no training data is available, the algorithm predicts a value of zero, because the offset is not taken into account during the training process. However, Gaussian process regression has used 6 iterations to approximate the minimum energy path and needs 9 iterations additionally to obtain the transition state.



(a) Müller-Brown surface (b) Fitted surface

Figure 3.7.: Comparing the Müller-Brown surface with the predicted landscape of Gaussian process regression after locating the transition state. The left figure (a) shows the minimum energy path approximation obtained by applying the nudged elastic band method directly to the surface and by using the machine learning procedure. The right figure (b) illustrates the predicted surface of the machine learning algorithm after locating the transition state can be seen. There are only those training points are drawn which are used to approximate the minimum energy path.

Iterative re-weighted least squares

This approach is an implementation of a support vector machine with an insensitive region ϵ around the predicted curve. Hence, it introduces a tolerated error, see Section 2.2.2. The training process of the iterative re-weighted least squares method relies on matrix inversion in each iteration and it stops if the Lagrangian is minimised and changes of it are below a given threshold. This makes the iterative re-weighted least squares method computationally more expensive then the other two approaches. However, during the training process it disposes training points containing redundant information about the problem. As a consequence, predictions are computationally

cheaper compared to the other two kernel methods. Furthermore, the iterative re-weighted least squares method uses a constant kernel hyper-parameter. Similar to the regularised least squares method the optimisation can be done via a grid search. Also for the iterative re-weighted least squares method a non-optimal hyper-parameter has only a minor effect on the stability of the algorithm, but affects the representation quality. Furthermore, iterative re-weighted least squares method is sensitive to high regularisation values, because they increase the error contribution in the Lagrangian which rises the probability of oscillations of the Lagrangian. It should be noted for the transition state search that the introduced error is smaller than the convergence criteria for the transition state search itself. The machine learning parameters for the iterative re-weighted least squares approach are summarised in Table 3.6.

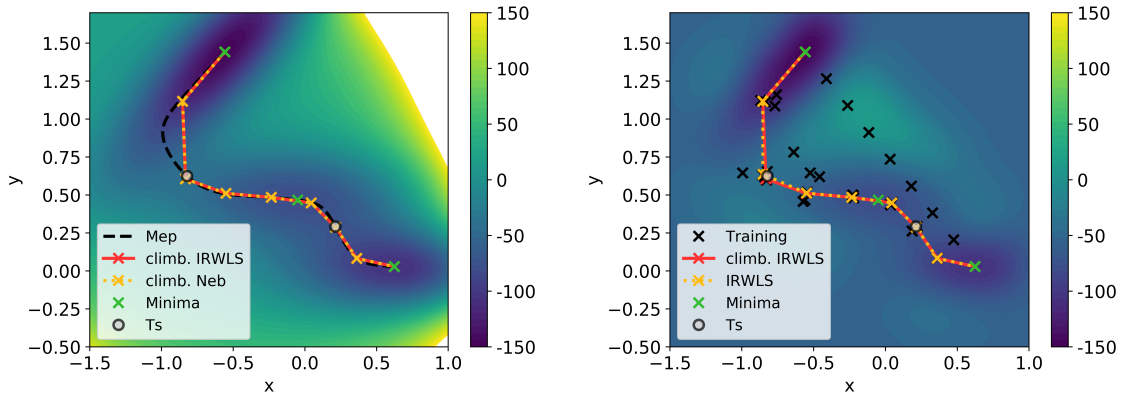
Figure 3.8 shows a comparison between the predicted fictitious landscape of the iterative re-weighted least squares approach and the Müller-Brown surface. It can be seen from Figure 3.8a that the minimum energy path as well as the transition state are well approximated and that a path similar to the plain nudged elastic band method is obtained. The training points to approximate the minimum energy path are drawn in Figure 3.8b. All further training points that are used to localise the transition state are not shown, because they are hardly distinguishable.

By comparing both potential energy surfaces it is obvious that the fictitious predicted landscape represents the Müller-Brown surface very well near the reaction path. At the outside, where the algorithm has no information about the real surface, it predicts a constant value. The minimum energy path was approximated by the iterative re-weighted least squares method within 5 iterations and it took 8 further iterations to locate the transition state.

Kernel method comparison

All three machine learning techniques have found the minimum energy path as well as the transition state reasonably and the obtained pathway is similar to the plain nudged elastic band method. Furthermore, the vicinity of the minimum energy path is well represented by all kernel methods, although the choice of training points differs significantly. In regions where the algorithms have no information about the Müller-Brown surface they predict a constant value. To improve the landscape in unknown territories further training points in these regions are needed. Similar to above this shows that the algorithms are able to interpolate very well, but extrapolation capabilities are very limited.

All three algorithms use a similar kernel hyper-parameter which defines the width for the radial basis functions. Thus, it is assumed that this parameter is a property of the surface. Tests with a larger hyper-parameter ($\sigma = 1.0$) have shown that the machine



(a) Müller-Brown surface

(b) Fitted surface

Figure 3.8.: Comparing the Müller-Brown surface with the fictitious predicted surface by the iterative re-weighted least squares technique after it has located the transition state. The left figure (a) shows the minimum energy path obtained by applying directly the nudged elastic band to the Müller-Brown surface and by using the machine learning procedure. The right figure (b) illustrates the predicted landscape via the iterative re-weighted least squares technique after locating the transition state.

learning algorithms have difficulties representing the Müller-Brown surface. A reason could be that the landscape features are small compared to the width of the radial basis function. On the other hand, a smaller hyper-parameter ($\sigma = 0.1$) slows down the convergence of the transition state search process, but it is still possible to represent the landscape. Additionally, it has revealed that using lower convergence criteria for the nudged elastic band on the predicted fictitious landscape than for convergence on the Müller-Brown surface does not improve the convergence speed. Maybe this is due to the rapid function value change on the Müller-Brown surface.

Figure 3.9 shows the minimum energy path approximations of all three kernel methods. It can be seen that all methods provide a similar path and locate the same transition state. As seen in the Figures 3.6a, 3.7a and 3.8a this path is the same path as the plain nudged elastic band method locates. In comparison to a direct plain transition state search, all machine learning techniques investigated in this thesis reduced the number of calculations on the Müller-Brown surface. This suggests that machine learning techniques are a highly useful tool to accelerate the transition state search.

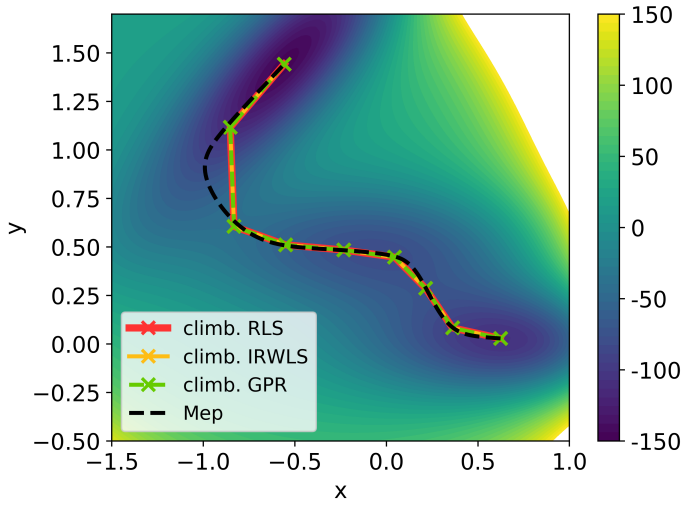


Figure 3.9.: Comparing the obtained minimum energy path after locating the transition state of all three machine learning algorithms with the real minimum energy path.

3.3. Molecular test systems

In this section the machine learning algorithms are tested on more realistic systems. For that purpose two molecules are selected, which have well known transition states and represent different structural transitions. These molecular systems are tabulated in Table 3.7 and their equilibrium structures are illustrated in Figure 3.10 (ammonia) and 3.11 (ethane).

Table 3.7.: Molecular test systems

Molecule	chem. formula	chemical reaction	Exchange	Basis set
ammonia	NH_3	Geometry inversion	Hartree-Fock	aug-cc-pvdz [36][37]
ethane	C_2H_6	CH_3 group rotation	Hartree-Fock	STO-3G [38]

All *ab initio* calculations are done with the QChem [39] package at the Hartree-Fock level of theory. The suggested method for transition state searches with QChem utilises the freezing string technique to approximate the transition state and applies a local gradient-following technique to locate it. However, for the sake of a direct comparison, a plain nudged elastic band method has been coded and is directly applied to the molecular potential energy surface as provided by the QChem routines. In the following, the pre-optimised path refers to the path that is obtained after the nudged elastic band method is used with the image-dependent pair potential.

Every machine learning method uses the same parameter for each molecular test system. Table 3.8 lists all parameters for each kernel method, except the kernel hyper-parameter which is assumed to be a surface parameter. Thus, it is given for each molecular test system separately. The neural network approach uses the parameter set of Peterson [1] and Artrith [40]. The network structure consists of two hidden layers with five neurons each. The hyperbolic tangent is used as activation function. Details can be found in the supplementary material of Peterson [1] and Artrith [40].

Table 3.8.: Machine learning parameters for the molecular systems

method	value reg.	grad. reg.	restarts	norm. y	ϵ	ϵ'	eps
RLS	10^{10}	10^{10}	-	-	-	-	-
GPR	10^{10}	10^{10}	5	No	-		-
IRWLS	10^{7}	10^{7}	-	-	10^{-5}	10^{-5}	10^{-5}

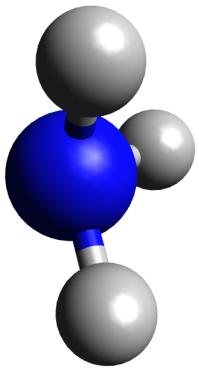


Figure 3.10.: Molecular structure of ammonia

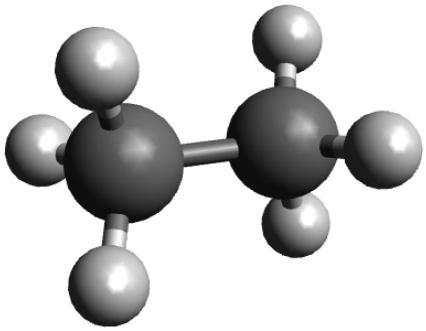


Figure 3.11.: Molecular structure of ethane

3.3.1. Ammonia

The investigated structural rearrangement of ammonia is an inversion. The nitrogen passes through the hydrogen plane to the other side. During this process, the bond length between the two atom species is compressed, see Figure 3.12. This leads to a high energy barrier which is on the order of bond breaking. Since the gradient component parallel to the band is high compared to the gradient component perpendicular to it, the images tend to slide down towards the minima with the mathematically-motivated tangent estimation. This happens even if a high spring constant value is chosen. Thus, the energy-weighted tangent estimation is used here.

In Table 3.9 all nudged elastic band parameters are summarised. The optimised kernel hyper-parameter for Gaussian process regression is $\sigma = 0.565$ for a linear initial path

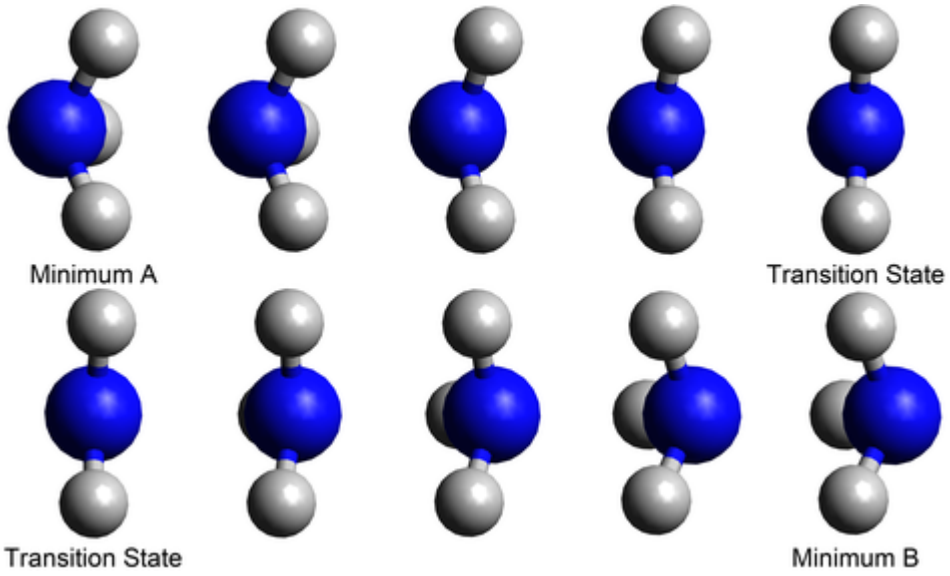


Figure 3.12.: Ammonia inversion geometries obtained by using the nudged elastic band method.

guess and $\sigma = 1.18$ for the pre-optimised initial pathway. The regularised least squares and iterative re-weighted least squares technique use a kernel hyper-parameter value of $\sigma = 0.4$. However, it is assumed that the kernel hyper-parameter is specific for every surface. This deviation might stem from the fact that Gaussian process regression does not fit the offset during the weight optimisation process.

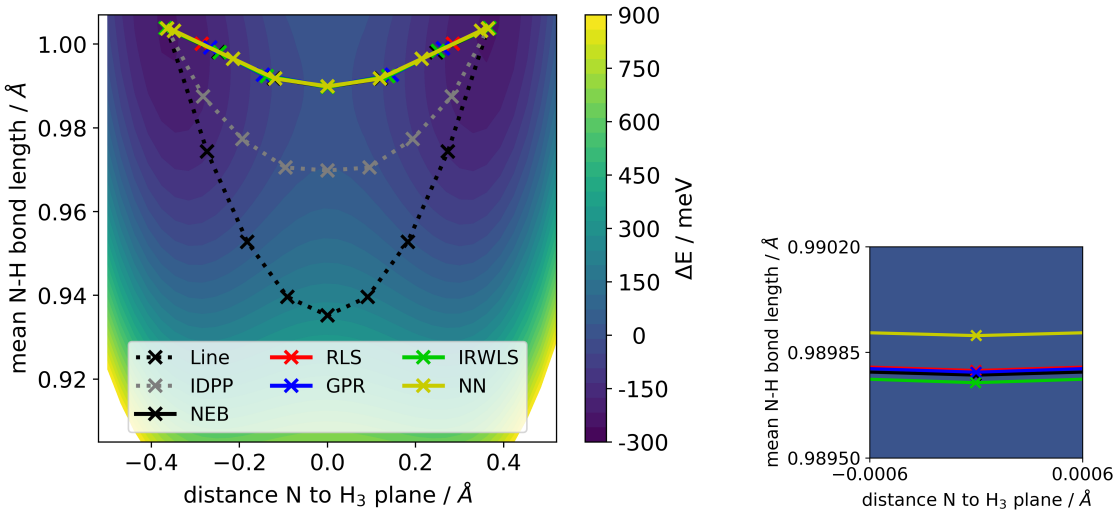
Table 3.9.: Nudged elastic band parameters used for the transition state search calculation of the inversion of ammonia.

Search	tangent method	ΔT	ΔT_{max}	r_{trust}	spring	conv. grad
IDPP	energy weighted	3.5	7.0	0.05	0.001	0.005
MEP	energy weighted	3.5	7.0	0.05	0.0001	0.001
TS	energy weighted	3.5	7.0	0.05	0.0001	0.0005

Since the atoms do not collide on the linearly interpolated path, there is no need to pre-optimise the latter with the image-dependent pair potential, which allows to use both paths for the tests. As before, all algorithms approximate the minimum energy path first, and then localise the transition state starting from the approximated reaction path.

By choosing convenient cuts through the potential energy surface it is possible to compare the calculated minimum energy paths visually; see Figure 3.13. For that, two intrinsic coordinates are defined, the distance between nitrogen atom and the hydrogen

plane, and the mean bond length between nitrogen and hydrogen. In Figure 3.13a it can be seen that the linearly interpolated path deviates significantly from the actual minimum energy path. Even the pre-optimised path is off the real reaction path, but is at least somewhat closer than the linearly interpolated initial pathway. All machine learning techniques obtain a similar minimum energy path as the plain nudged elastic band method does.



(a) Overview (b) Transition states

Figure 3.13.: Cut through the potential energy surface of ammonia with the intrinsic coordinates, mean bond length of H to N as y-axis and the distance from N to the H_3 plane as x-axis. As energy reference is the QChem calculated transition state used, with -1529.21 eV.

Only a very close look at the transition state reveals some minimal numerical differences, see Figure 3.13b. For comparison, the transition state properties are calculated and listed in Table 3.10 and 3.11. There it can be seen that all transition states are close to the reference state calculated with QChem. All machine learning techniques have reduced the number of *ab initio* calls significantly, in particular the neural network approach, which seems superior to the kernel methods in the tested setup. This was expected since the neural network takes molecular symmetries such as rotational and translational invariances into account. Hence, the neural network does not have to learn these invariances but can use them right away. Since the molecular potential energy surface for that transition is symmetric this reduces the number of *ab initio* calls significantly. Interestingly, by looking at the plain nudged elastic band method calculation, one can see that the linear path needs less *ab initio* calls than the pre-optimised pathway to converge. This is a very unusual behaviour since the initial optimised guess is closer to the actual reaction path, see Figure 3.13a. However, one explanation might be that the perpendicular gradient component of the linearly

interpolated guess is large compared to the pre-optimised band, which makes the optimisation method take smaller steps in case of the pre-optimised initial guess.

Table 3.10.: Transition state properties of ammonia for the linearly interpolated path. As reference, the calculated values of QChem are used with an energy of -1529.21 eV and a frequency of -873.39 cm^{-1}.

Method	ΔEnergy /μeV	Frequency / cm^{-1}	*ab initio* calls	
			MEP	TS
NEB	8.882	-872.80	149	168
RLS	9.102	-872.67	37	7
GPR	8.378	-872.72	44	14
IRWLS	8.444	-873.02	51	7
NN	-14.909	-871.67	16	7

Table 3.11.: Calculated transition state properties of ammonia for the pre-optimised path guess. As reference the calculated values of QChem are used with an energy of -1529.21 eV and a frequency of -873.39 cm^{-1}.

Method	ΔEnergy /μeV	Frequency / cm^{-1}	*ab initio* calls	
			MEP	TS
NEB	215.332	-871.96	282	196
RLS	9.600	-874.93	23	7
GPR	9.611	-871.97	37	7
IRWLS	9.216	-872.24	44	14
NN	-243.427	-871.98	16	7

3.3.2. Ethane

Ethane is a very common test system for transition state searches. The investigated structural rearrangement is given by the rotation of both methyl groups (CH_3). One rotates clockwise, while the other one rotates counterclockwise. Both rotate by 60°, as illustrated in Figure 3.14. On the minimum energy path the bond lengths change slightly, which leads to a low barrier height.

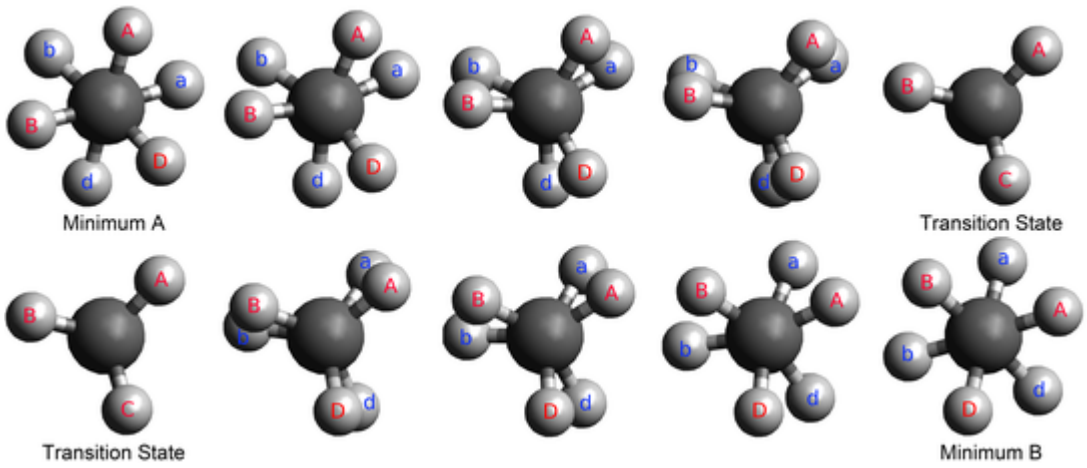


Figure 3.14.: Geometries of the ethane rotation obtained by using the nudged elastic band method.

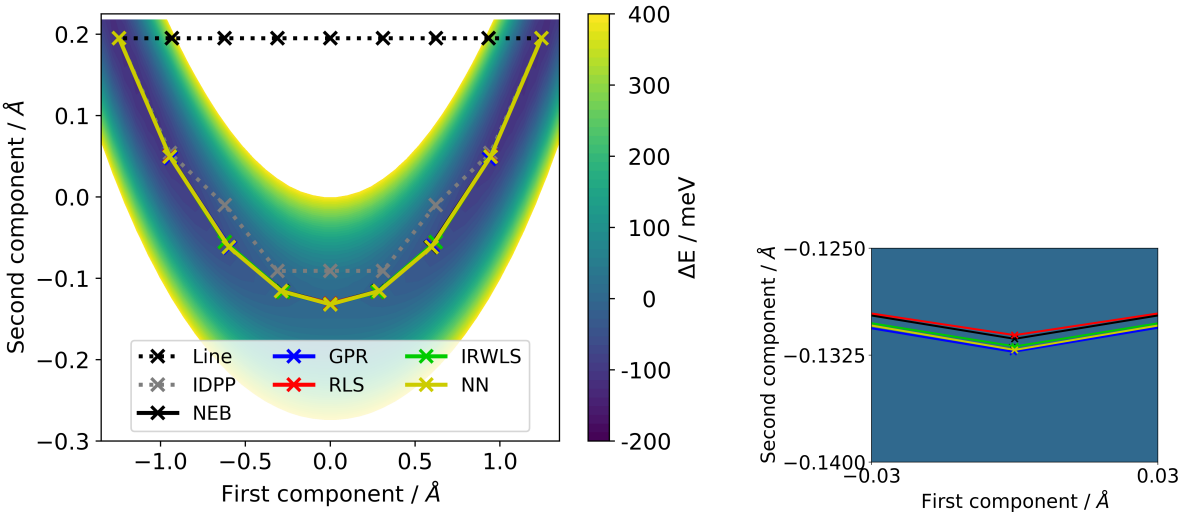
All parameters for the nudged elastic band calculation are listed in Table 3.12. The kernel hyper-parameter was set to $\sigma = 0.4$ for the iterative re-weighted least squares and regularised least squares approach. Gaussian process regression optimises the hyper-parameter value to $\sigma = 1.62$ for the linearly interpolated initial path guess and to value of $\sigma = 3.8$ when starting from the pre-optimised path as initial guess.

Table 3.12.: Nudged elastic band parameters used for the transition state search calculation of the methyl group rotation of ethane.

Search	tangent method	ΔT	ΔT_{max}	r_{trust}	spring	conv. grad
MEP	math. motivated	3.5	7.0	0.05	0.0001	0.001
TS	math. motivated	3.5	7.0	0.05	0.0001	0.0005

Since atoms do not collide on the linearly interpolated initial pathway, it is possible to use it for testing as well as the pre-optimised path. This reaction has more degrees of freedom, hence, it is hard to define two intrinsic coordinates that represent the transition from one minimum to the other. Hence, the principle component analysis of SciKit-Learn [41] is utilised to define two coordinates that allow for a convenient

cut through the potential energy surface, see Figure 3.15. Note that each coordinate describes a molecular motion of several atoms. In order to get a higher energy resolution around the reaction path all energy values above 375 meV are neglected in the figure. Figure 3.15a compares the different minimum energy paths. It is obvious to see that the linear initial band guess lies far from the actual reaction path, while the pre-optimised linear path is already close to the final minimum energy path. Moreover, all machine learning techniques have approximated a reaction path that is similar to the obtained pathway with the plain nudged elastic band method.



(a) Overview (b) Transition states

Figure 3.15.: Cut through the potential energy surface of ethane by using a principle component analysis. As energy reference is the calculated transition state energy of QChem used which is -2130.69 eV.

The obtained transition state energies show only minimal numerical variations as it can be seen in Figure 3.15b. Thus, the transition state properties are calculated and compared to the QChem reference, see Table 3.13 and 3.14. This comparison shows that all transition states are well located. Looking at the number of *ab initio* calls, it is obvious to see that all machine learning techniques reduce the number of calls in case of the linear initial band guess. Also in case of the pre-optimised initial path these methods are able to reduce the amount of *ab initio* calls. Again, the neural network approach is superior to the kernel methods, most likely due to the fact that it takes molecular symmetries into account. The iterative re-weighted least squares approach showed some convergence problems in case of the pre-optimised initial path, where it got stuck in the weight-optimisation process and had to be terminated after a few days of calculation time. This issue might have been caused by oscillations of the Lagrangian due to small variations of the energy values.

Table 3.13.: Transition state properties of ethane for the linearly interpolated initial path. The transition state obtained with QChem is used as reference with an energy of -2130.69 eV and a frequency of -309.48 cm^{-1}.

Methods	Δ Energy / μeV	Frequency / cm^{-1}	*ab initio* steps	
			MEP	Ts
NEB	-51.394	-312.21	275	245
RLS	-147.820	-311.68	72	14
GPR	-27.565	-309.03	72	7
IRWLS	-14.659	-312.61	65	7
NN	-78.306	-313.62	16	7

Table 3.14.: Transition state properties of ethane for the pre-optimised initial path. The obtained transition state by QChem is used as reference with an energy of -2130.69 eV and a frequency of -309.48 cm^{-1}.

Methods	Δ Energy / μeV	Frequency / cm^{-1}	*ab initio* steps	
			MEP	Ts
NEB	-59.498	-308.56	219	168
RLS	-48.344	-310.24	44	14
GPR	-28.703	-309.88	37	7
NN	-27.505	-302.64	16	7

4. Conclusion and Outlook

In this thesis different machine learning techniques were investigated which are able to accelerate transition state searches based on the nudged elastic band method. A neural network approach and three kernel methods were implemented. First, these techniques were tested for their capability of approximating the minimum energy path on the two-dimensional Müller-Brown surface. Second, all machine learning methods were used to approximate reaction paths and to locate transition states in two molecular systems with well known reaction paths. The number of necessary *ab initio* calls was compared to that obtained with the plain nudged elastic band method; in this case also for different initial pathways.

In the first test only the kernel methods were used exclusively, since the neural network approach transforms the input values with descriptors operating in three-dimensional space in order to take molecular symmetries into account. Thus, the latter can be only applied to molecules. All kernel methods were capable of approximating the minimum energy path and of locating the transition state as well as the plain nudged elastic band method does. The tests further showed that the kernel methods were able to represent the vicinity of the reaction path very well, which suggests large benefits for structure evaluations for example in the course of molecular dynamics simulations on an *ab initio* surface. Interestingly, all kernel methods use a similar kernel hyper-parameter. Thus, it is suspected that this parameter is a property mostly determined by the potential energy surface. Additionally, all kernel methods reduce the number of evaluations on the Müller-Brown surface needed to locate the transition state.

In a second test, the molecular systems ammonia and ethane have been selected. In the case of a linearly interpolated initial band guess all machine learning algorithms were capable of locating the transition state within chemical accuracy. A similar result could be found when using the pre-optimised initial band path, except for the iterative re-weighted least squares method, which had some difficulties with weight-optimisation for ethane. This might have been caused by the initial path being close to the actual reaction path and the minimal changes during update. The algorithm got stuck in the weight-optimisation process due to an oscillation of the Lagrangian and the calculation was stopped manually after a few days. Thus, it may be useful to implement a sequential minimisation method [42][43] which reduces the computational burden of the weight-optimisation since it does not rely on a matrix inversion.

Furthermore, the molecular tests have shown that the optimised kernel hyper-parameter of Gaussian process regression differs significantly from the two other kernel method

approaches. This might be due to the fact that Gaussian process regression did not fit the offset during weight calculation.

Compared to the plain nudged elastic band method the number of *ab initio* calls was significantly reduced by all machine learning techniques. The neural network had the highest impact on the evaluation number, most likely because it makes use of molecular symmetries, which seems to make this technique superior to the kernel methods in the current setup.

From this it can be concluded that also kernel methods would benefit significantly from similar information. The smooth overlap of atomic positions kernel [44] might be a useful ansatz for a future implementation. Future studies of reaction pathways need to be extended towards bond-breaking, which has not been treated within this work. Particularly interesting in this context are reactions of molecules in gas phase on metallic surfaces, where the calculation of realistic potential energy surfaces is very costly but needed to estimate catalytic activities and selectivities.

A. Mathematical Derivations

A.1. Born-Oppenheimer Approximation

The Born-Oppenheimer approximation is an essential approach in quantum chemistry. It decouples the electron motion from the nuclei motion and assumes that the total molecular wave function can be written as a product of electron and nuclei wave functions. Qualitatively this is allowed due to the fact that the mass ratio between electron and nuclei is $\frac{m_{\text{nuclei}}}{\text{m}_{\text{elec}}} \approx 1835$. The following derivations follow roughly the book of Combs [1].

The approximation starts with the molecular Schrödinger equation

$$\mathcal{H} \left|\psi(\boldsymbol{R}, \boldsymbol{r})\right\rangle = E \left|\psi(\boldsymbol{R}, \boldsymbol{r})\right\rangle, \tag{A.1}$$

using the molecular Hamiltonian in atomic units, which is given by

$$\begin{aligned} \mathcal{H} = &- \underbrace{\sum_{i=1}^{N_{\text{elec}}} \frac{1}{2} \boldsymbol{\nabla}_i^2}_{T_{\text{e}}} - \underbrace{\sum_{A=1}^{N_{\text{nuclei}}} \frac{1}{2M_A} \boldsymbol{\nabla}_A^2}_{T_{\text{N}}} - \underbrace{\sum_{i=1}^{N_{\text{elec}}} \sum_{A=1}^{N_{\text{nuclei}}} \frac{Z_A}{|\boldsymbol{R}_A - \boldsymbol{r}_i|}}_{V_{\text{eN}}} \\ &+ \underbrace{\sum_{i=1}^{N_{\text{elec}}} \sum_{j>i}^{N_{\text{elec}}} \frac{1}{|\boldsymbol{r}_i - \boldsymbol{r}_j|}}_{V_{\text{ee}}} + \underbrace{\sum_{A=1}^{N_{\text{nuclei}}} \sum_{B>A}^{N_{\text{nuclei}}} \frac{Z_A Z_B}{|\boldsymbol{R}_A - \boldsymbol{R}_B|}}_{V_{\text{NN}}}, \end{aligned} \tag{A.2}$$

with T_{e} as kinetic energy of the electrons, T_{N} as kinetic energy of the nuclei, V_{eN} as the electron-nuclei interaction potential, V_{ee} as the electron-electron interaction potential and V_{NN} as the nuclei-nuclei interaction potential. In the molecular Hamiltonian upper case letters denote nuclei variables and the lower case letters electron variables. The Born-Oppenheimer approximation assumes that the nuclei are at rest and do not move within the calculation period. Thus, the nuclei coordinates $\boldsymbol{R}$ turn into a parameter and the nuclei kinetic energy vanishes from the Hamiltonian. This leads to the electronic Schrödinger equation given by

$$\left(-T_{\text{e}} - V_{\text{eN}} + V_{\text{ee}} + V_{\text{NN}} - \epsilon_m(\boldsymbol{R})\right) \psi(\boldsymbol{R}, \boldsymbol{r}) = 0, \tag{A.3}$$

[1] Proven in [45] page 1 et sequentes.

where ϵ_m denotes the eigen energy of the m-th electron. In addition, the total solution of the Schrödinger equation is expanded into electronic coordinates which leads to

$$\psi(\boldsymbol{R},\boldsymbol{r}) = \sum_{m=0}^{\infty} \xi_m(\boldsymbol{R})\phi_m(\boldsymbol{R},\boldsymbol{r}). \tag{A.4}$$

Through that, the Schrödinger equation can be written as

$$(-T_\mathrm{e} - T_\mathrm{N} - V_\mathrm{Ne} + V_\mathrm{ee} + V_\mathrm{NN}) \sum_{m=0}^{\infty} \xi_m(\boldsymbol{R})\phi_m(\boldsymbol{R},\boldsymbol{r}) = E \sum_{m=0}^{\infty} \xi_m(\boldsymbol{R})\phi_m(\boldsymbol{R},\boldsymbol{r}). \tag{A.5}$$

Multiplying this from the left by $\phi_n^*(\boldsymbol{R},\boldsymbol{r})$ and integrating over the electron coordinates $\boldsymbol{r}$ gives

$$\sum_{m=0}^{\infty} \int \mathrm{d}\boldsymbol{r}\phi_n^*(\boldsymbol{R},\boldsymbol{r}) \left(-T_\mathrm{e} - T_\mathrm{N} - V_\mathrm{eN} + V_\mathrm{ee} + V_\mathrm{NN} - E\right) \xi(\boldsymbol{R})\phi(\boldsymbol{R},\boldsymbol{r}) = 0. \tag{A.6}$$

With the orthogonality condition $\int \mathrm{d}\boldsymbol{r}\phi_n^*(\boldsymbol{R},\boldsymbol{r})\phi_m(\boldsymbol{R},\boldsymbol{r}) = \delta_{nm}$ and by using Equation A.3 this equation can be expressed as

$$\sum_{m=0}^{\infty} \left(\int \mathrm{d}\boldsymbol{r}\phi_n^*(\boldsymbol{R},\boldsymbol{r})(-T_\mathrm{N})\phi(\boldsymbol{R},\boldsymbol{r}) + (\epsilon_m(\boldsymbol{R}) - E)\delta_{nm} \right) \xi(\boldsymbol{R}) = 0. \tag{A.7}$$

The first part of this equation contains the differential operator T_N. Applying this operator to the two wave functions $\xi(\boldsymbol{R})$ and $\phi(\boldsymbol{R},\boldsymbol{r})$ leads to

$$\begin{aligned}
\int \mathrm{d}\boldsymbol{r}\phi_n^*(\boldsymbol{R},\boldsymbol{r})T_\mathrm{N}\phi_m(\boldsymbol{R},\boldsymbol{r})\xi_m = &\sum_{A=1}^{N_\mathrm{nuclei}} \frac{1}{2M_A} \sum_{d=1}^{3} \int \mathrm{d}\boldsymbol{r}\phi_n^*(\boldsymbol{R},\boldsymbol{r}) \frac{\partial^2\phi_m(\boldsymbol{R},\boldsymbol{r})}{\partial R_{A,d}^2} \xi_m(\boldsymbol{R}) \\
&+ 2 \sum_{A=1}^{N_\mathrm{nuclei}} \frac{1}{2M_A} \sum_{d=1}^{3} \int \mathrm{d}\boldsymbol{r}\phi_n^* \frac{\partial\phi_m(\boldsymbol{R},\boldsymbol{r})}{\partial R_{A,d}} \frac{\partial\xi_m(\boldsymbol{R})}{\partial R_d} \\
&+ \underbrace{\sum_{A=1}^{N_\mathrm{nuclei}} \frac{1}{2M_A} \sum_{d=1}^{3} \frac{\partial^2\xi_m(\boldsymbol{R})}{\partial R_d^2} \delta_{nm}}_{T_\mathrm{N}\xi_m\delta_{mn}}
\end{aligned} \tag{A.8}$$

Plugging this into Equation A.7 leads to a coupled differential equations system, which

is given by

$$\Big[\sum_{A=1}^{N_{\text{nuclei}}}\frac{1}{2M_A}\sum_{d=1}^{3}\frac{\partial^2}{\partial R_{A,d}^2}+\epsilon_m-E\Big]\xi_n(\boldsymbol{R})= \\ \sum_{m}^{\infty}\sum_{A=1}^{N_{\text{nuclei}}}\frac{1}{M_A}\sum_{d=1}^{3}\int \mathrm{d}\boldsymbol{r}\phi_n^*\frac{\partial\phi_m(\boldsymbol{R},\boldsymbol{r})}{\partial R_{A,d}}\frac{\partial\xi_m(\boldsymbol{R})}{\partial R_{A,d}}+ \\ \sum_{m}^{\infty}\sum_{A=1}^{N_{\text{nuclei}}}\frac{1}{2M_A}\sum_{d=1}^{3}\int \mathrm{d}\boldsymbol{r}\phi_n^*(\boldsymbol{R},\boldsymbol{r})\frac{\partial^2\phi_m(\boldsymbol{R},\boldsymbol{r})}{\partial R_{A,d}}\xi_m(\boldsymbol{R}). \tag{A.9}$$

By defining coupling matrix elements

$$\Lambda_{mn}=\sum_{m}^{\infty}\sum_{A=1}^{N_{\text{nuclei}}}\frac{1}{2M_A}\sum_{d=1}^{3}\int \mathrm{d}\boldsymbol{r}\phi_n^*(\boldsymbol{R},\boldsymbol{r})\Big[2\big(\frac{\partial}{\partial R_{A,d}}\phi_m(\boldsymbol{R},\boldsymbol{r})\big)\frac{\partial}{\partial R_{A,d}}+ \\ \frac{\partial^2}{\partial R_{A,d}^2}\phi_m(\boldsymbol{R},\boldsymbol{r})\Big], \tag{A.10}$$

The coupled differential equation system can be written as

$$\left[T_{\text{N}}+\epsilon_n(\boldsymbol{R})-E\right]\xi_n(\boldsymbol{R})=\sum_{m}^{\infty}\Lambda_{mn}\xi_m(\boldsymbol{R}). \tag{A.11}$$

The Born-Oppenheimer approximation assumes that the electronic wave function $\phi(\boldsymbol{R},\boldsymbol{r})$ does not depend on the nuclei coordinates $\boldsymbol{R}$ as a variable, but as a parameter. Thus in this approximation the Λ_{nm} vanishes, leading to

$$\left[T_{\text{N}}+\epsilon_n(\boldsymbol{R})-E\right]\xi_n(\boldsymbol{R})=0 \tag{A.12}$$

$$\psi_{n\nu}(\boldsymbol{R},\boldsymbol{r})=\phi_n(\boldsymbol{R},\boldsymbol{r})\xi_\nu(\boldsymbol{R}), \tag{A.13}$$

where the total molecular wave function expansion reduces to one term. In order to be a valid assumption, the Born-Oppenheimer approximation has to satisfy the condition

$$\frac{\left|\left\langle\xi_{n\nu}\right|\Lambda_{nm}\left|\xi_{m\nu'}\right\rangle\right|}{|E_{n\nu}-E_{m\nu'}|}\ll 1 \qquad \text{for every} n\neq m,\nu\neq\nu', \tag{A.14}$$

which means that the energy difference has to be large between different states compared to the matrix elements. If the energy difference is on the same order of magnitude as vibronic frequencies the whole algorithm breaks down. Exceptions are the states where Λ_{nm} is zero for symmetry reasons.

A.2. Hohenberg-Kohn Theorem

The first theorem states that two different external potentials do not give the same density. We start from the molecular Hamiltonian which is given by

$$\begin{aligned}\mathcal{H}_{\text{elec}} = &-\sum_{i=1}^{N_{\text{elec}}} \frac{1}{2}\boldsymbol{\nabla} i^2 - \sum_{i=1}^{N_{\text{elec}}} \sum_{A=1}^{N_{\text{nuclei}}} \frac{Z_A}{|\boldsymbol{R}_A - \boldsymbol{r}_i|} \\ &+ \sum_{i=1}^{N_{\text{elec}}} \sum_{j>i}^{N_{\text{elec}}} \frac{1}{|\boldsymbol{r}_i - \boldsymbol{r}_j|} + \sum_{A=1}^{N_{\text{nuclei}}} \sum_{B>A}^{N_{\text{nuclei}}} \frac{Z_A Z_B}{|\boldsymbol{R}_A - \boldsymbol{R}_B|}.\end{aligned} \tag{A.15}$$

The proof consists of a comparison between two Hamiltonians with different eigenfunctions and eigenvalues which are given by

$$\langle\psi|\,\mathcal{H}\,|\psi\rangle = E_0, \tag{A.16}$$
$$\langle\psi'|\,\mathcal{H}'\,|\psi'\rangle = E_0'. \tag{A.17}$$

Applying a different wave function to the Hamiltonian leads to an eigenvalue that is higher than the ground state energy, so are obtaining the inequalities

$$E_0 < \langle\psi'|\,\mathcal{H}\,|\psi'\rangle = \langle\psi'|\,\mathcal{H}'\,|\psi'\rangle + \langle\psi'|\,\mathcal{H} - \mathcal{H}'\,|\psi'\rangle\,, \tag{A.18}$$
$$E_0' < \langle\psi|\,\mathcal{H}'\,|\psi\rangle = \langle\psi|\,\mathcal{H}\,|\psi\rangle + \langle\psi|\,\mathcal{H}' - \mathcal{H}\,|\psi\rangle\,. \tag{A.19}$$

Combining these two lines leads to

$$E_0 + E_0' < E_0' + \langle\psi'|\,\mathcal{H} - \mathcal{H}'\,|\psi'\rangle + E_0 + \langle\psi|\,\mathcal{H}' - \mathcal{H}\,|\psi\rangle\,, \tag{A.20}$$

where the ground state definition of Equation A.17 has been used. The two other terms on the right can be rewritten as an integral over the external potential. From that the proof can be completed as follows

$$E_0 + E_0' < E_0' + \int \mathrm{d}\boldsymbol{r}\rho(\boldsymbol{r})(V_{\text{ext}} - V_{\text{ext}}') + E_0 + \int \mathrm{d}\boldsymbol{r}\rho(\boldsymbol{r})(V_{\text{ext}}' - V_{\text{ext}}), \tag{A.21}$$

which leads to

$$E_0 + E_0' < E_0' + E_0. \tag{A.22}$$

Since such an inequality is not possible this enforces that an external potential is uniquely connected with the electron density.

A.3. Tangent estimation to the nudged elastic band

Improved nudged elastic band calculations are based on the estimation of the tangent for each image. The simplest case is to use the direction of the neighbouring images, given by

$$\hat{\boldsymbol{\tau}}_i = \frac{\boldsymbol{x}_{i+1} - \boldsymbol{x}_{i-1}}{|\boldsymbol{x}_{i+1} - \boldsymbol{x}_{i-1}|}, \tag{A.23}$$

where the subscript i defines image of which the tangent is calculated. This equation can be improved by including also the image in between. The tangent for the image in the middle is calculated by

$$\hat{\boldsymbol{\tau}}_i = \frac{\boldsymbol{x}_{i+1} - \boldsymbol{x}_i}{|\boldsymbol{x}_{i+1} - \boldsymbol{x}_i|} + \frac{\boldsymbol{x}_i - \boldsymbol{x}_{i-1}}{|\boldsymbol{x}_i - \boldsymbol{x}_{i-1}|}. \tag{A.24}$$

Here, it should be noted that the tangent must be of unit length. Therefore, all of them have to be normalised before applying them in the nudged elastic band procedure. However, the two mentioned tangent estimations are mathematically-motivated and in some cases problematic, especially if the path is very kinky. Another approach, physically motivated, is to weight the tangent with the energy of neighbouring images [11]. For this estimation auxiliary tangents are used:

$$\begin{aligned} \hat{\boldsymbol{\tau}}_i^+ &= \boldsymbol{x}_{i+1} - \boldsymbol{x}_i, \\ \hat{\boldsymbol{\tau}}_i^- &= \boldsymbol{x}_i - \boldsymbol{x}_{i-1}. \end{aligned} \tag{A.25}$$

The estimated tangent to the band is then given by

$$\hat{\boldsymbol{\tau}}_i = \begin{cases} \hat{\boldsymbol{\tau}}_i^+ & \text{if } E_{i+1} > E_i > E_{i-1} \\ \hat{\boldsymbol{\tau}}_i^- & \text{if } E_{i+1} < E_i < E_{i-1} \\ \hat{\boldsymbol{\tau}}_i^+ \Delta E_i^{\max} + \hat{\boldsymbol{\tau}}_i^- \Delta E_i^{\min} & \text{if } E_{i+1} > E_{i-1} \\ \hat{\boldsymbol{\tau}}_\mathrm{i}^+ \Delta E_i^{\min} + \hat{\boldsymbol{\tau}}_i^- \Delta E_i^{\max} & \text{if } E_{i+1} < E_{i-1} \end{cases}. \tag{A.26}$$

Here the energy value is used for a smooth switching between the two auxiliary tangents. The weight is calculated from the energy via

$$\begin{aligned} \Delta E_i^{\max} &= \max(E_{i+1} - E_i, E_{i-1} - E_i), \\ \Delta E_i^{\min} &= \min(E_{i+1} - E_i, E_{i-1} - E_i). \end{aligned} \tag{A.27}$$

Through this, the energy-weighted tangent can be calculated. However, it should be noted that the force projection is no longer exactly aligned with the band direction

and some positive effects, which are employed by introducing tangents to the nudged elastic band formalism, are neglected.

A.4. Image-Dependent Pair Potential

The image-dependent pair potential was proposed by Smidstrup et al. [14]. This method improves the initial band guess for the nudged elastic band. It has a lot in common with the linear synchronous transit method proposed by Halgren et al. [46], but with the advantage that it produces a continuous path and is more robust.

To start with the image-dependent pair potential all pair distances for each atom in both minimum geometries have to be calculated. Geometries are given in Cartesian coordinates and the direction can be evaluated with the Euclidean metric. Through this, an ideal distance for each intermediate image can be calculated [14]:

$$d_{i,j}^{\kappa} = d_{i,j}^{\alpha} + \kappa \frac{d_{i,j}^{\beta} - d_{i,j}^{\alpha}}{N}, \tag{A.28}$$

where $d_{i,j}^{\kappa}$ is the distance in image κ from atom i to atom j. α and β denote the two minimum geometries and N is the number of images. The objective function for the image-dependent pair potential can be stated as

$$S_{\mathrm{idpp}}^{\kappa} = \sum_{i=1}^{N} \sum_{j>i}^{N} w(d_{i,j}) \left(d_{i,j}^{\kappa} - |\boldsymbol{x}_i - \boldsymbol{x}_j| \right)^2, \tag{A.29}$$

with w as a weight function that represents a penalty term for too short distances. Minimisation of this function by variation in x directs the atoms into a configuration where all distances are close to the interpolated distances of Equation A.28. This optimisation can be achieved by applying the nudged elastic band algorithm which requires knowledge of the gradient of the objective function.

$$F_{\mathrm{idpp}}^{\kappa} = (\nabla_i w(d_{i,j}))|\boldsymbol{x}_i - \boldsymbol{x}_j| + w(d_{\mathrm{i,j}}) \frac{2(\boldsymbol{x}_i - \boldsymbol{x}_j)(d_{i,j}^{\kappa} - |\boldsymbol{x}_i - \boldsymbol{x}_j|)}{|\boldsymbol{x}_i - \boldsymbol{x}_j|}. \tag{A.30}$$

With this equation it is possible to improve the nudged elastic band initial guess for the real calculation. However, it should be noted, that the initial guess for this improvement is still the linearly interpolated in Cartesian coordinates.

A.5. Primal To Dual

The primal optimisation problem is given as a function called $f(x)$ which has to match certain conditions. These are expressed as equalities $h_j(x)$ and inequalities $g_i(x)$ constraints. Optimisation can be restricted to minimisation since it is possible to obtain maximisation simply by changing the sign of the objective function [2]. The primal problem can then be stated as

$$\begin{aligned} &\text{optimise } f(x) \\ &\text{subjected to } \begin{cases} g_i(x) \leq 0 & i = 1, \ldots, p. \\ h_j(x) = 0 & j = 1, \ldots, q \end{cases} \end{aligned} \tag{A.31}$$

First, the dual Lagrangian is constructed from the primal problem by including all constrains with Lagrangian multipliers into the objective function,

$$\begin{aligned} &\mathcal{L} = f(x) - \sum_{i=1}^{p} \lambda_i g_i(x) - \sum_{j=1}^{q} \mu_j g_j(x), \\ &\text{subjected to } \lambda_i, \mu_j \geq 0, \end{aligned} \tag{A.32}$$

where λ_i and μ_j are the dual variables. This equation contains primal and dual variables. It is possible to dispose the primal variables by minimising the Lagrangian with respect to them and eliminate subsequently all primal variables. This leads directly to the dual objective function, but in order to keep generality it is still stated as a function of both variables.

$$\begin{aligned} &\underset{x,\lambda,\mu}{\text{maximise}}\ f(x) + \sum_{i=1}^{p} \lambda_i g_i(x) + \sum_{j=1}^{q} \mu_j g_j(x), \\ &\text{subjected to } \lambda_i, \mu_j \geq 0. \end{aligned} \tag{A.33}$$

The solution is then obtained by maximising the dual objective function with respect to the Lagrangian multipliers. This is turned into a minimisation by changing the sign of $f(x)$ in equations A.32 and Equation A.33.

[2] A documentation about the Lagrangian multiplier can be found in [47], page 195 eq sequentes.

A.6. Karush-Kuhn-Tucker Condition

The Karush-Kuhn-Tucker (KKT) condition [29] is applied to solve constrained optimisation problems. It uses the dual Lagrangian, see Equation A.32. However, note that the condition is only valid if all functions are differentiable. The condition allows to calculate the optimal solution to the primal and dual problem. Therefore the Lagrangian is derived with respect to each primal and dual variable. This leads to a set of equations

$$\frac{\partial}{\partial x}\mathcal{L} \leq 0 \qquad \frac{\partial}{\partial \lambda_i}\mathcal{L} \leq 0 \qquad \frac{\partial}{\partial \mu_j}\mathcal{L} \leq 0 \qquad x, \lambda, \mu \geq 0. \tag{A.34}$$

The KKT condition is completed with the complementary slackness, which states that at the optimal solution all products between dual variables and constrains have to vanish, i. e.

$$\lambda_i g_i = 0, \tag{A.35}$$
$$\mu_j h_j = 0. \tag{A.36}$$

By substituting these equations into the original Lagrangian a set of equations is obtained. The solution to this set is also the optimal solution to the primal and dual problem.

A.7. Second-Order Taylor Expansion Of The Lagrangian

The iterative re-weighted least square (IRWLS) process, proposed by Lázaro et al. [26] uses a quadratic approximation to the Lagrangian, because each step in the IRWLS procedure relies on the previous solution. First the Lagrangian function is introduced.

$$\mathcal{L} = \frac{1}{2}\boldsymbol{w}^\intercal \boldsymbol{w} + C\sum_{i=1}^{N}(L(\xi_i) - L(\xi_i^*)) + D\sum_{i=1}^{N}(L(\tau_i) - L(\tau_i^*)) \tag{A.37}$$

which is similar to Equation 2.33. This expression gets modified with the first-order

Taylor expansion, which leads to

$$\begin{aligned}\mathcal{L}' = \frac{1}{2}\boldsymbol{w}^\intercal\boldsymbol{w} &+ C\left(\sum_{i=1}^{N} L(\xi_i^k) + \frac{dL(u)}{du}\Big|_{\xi_i^k}[\xi_i - \xi_i^k]\right) \\ &+ C\left(\sum_{i=1}^{N} L(\xi_i^{*k}) + \frac{dL(u)}{du}\Big|_{\xi_i^{*k}}[\xi_i^* - \xi_i^{*k}]\right) \\ &+ D\left(\sum_{i=1}^{N} L(\tau_i^k) + \frac{dL(u)}{du}\Big|_{\tau_i^k}[\tau_i - \tau_i^k]\right) \\ &+ D\left(\sum_{i=1}^{N} L(\tau_i^{*k}) + \frac{dL(u)}{du}\Big|_{\tau_i^{*k}}[\tau_i^* - \tau_i^{*k}]\right)\end{aligned}, \tag{A.38}$$

where $\mathcal{L}' = \mathcal{L}$ and $\nabla\mathcal{L}' = \nabla\mathcal{L}$. To obtain a quadratic approximation one has to take the second-order Taylor expansion. It has to be imposed that $\mathcal{L}'' = \mathcal{L}$ and $\nabla\mathcal{L}'' = \nabla\mathcal{L}$, so that the quadratic approximation is still a solution to the original problem. This leads to

$$\begin{aligned}\mathcal{L}'' = \frac{1}{2}\boldsymbol{w}^\intercal\boldsymbol{w} &+ C\left(\sum_{i=1}^{N} L(\xi_i^k) + \frac{dL(u)}{du}\Big|_{\xi_i^k}\frac{(\xi_i)^2 - (\xi_i^k)^2}{2\xi_i^k}\right) \\ &+ C\left(\sum_{i=1}^{N} L(\xi_i^{*k}) + \frac{dL(u)}{du}\Big|_{\xi_i^{*k}}\frac{(\xi_i^*)^2 - (\xi_i^{*k})^2}{2\xi_i^{*k}}\right) \\ &+ D\left(\sum_{i=1}^{N} L(\tau_i^k) + \frac{dL(u)}{du}\Big|_{\tau_i^k}\frac{(\tau_i)^2 - (\tau_i^k)^2}{2\tau_i^k}\right) \\ &+ D\left(\sum_{i=1}^{N} L(\tau_i^{*k}) + \frac{dL(u)}{du}\Big|_{\tau_i^{*k}}\frac{(\tau_i^*)^2 - (\tau_i^{*k})^2}{2\tau_i^{*k}}\right)\end{aligned}, \tag{A.39}$$

By using the definition for error weights

$$a_i = \frac{C}{\xi_i^k}\Big|_{\xi_i^k} \quad ,a_i^* = \frac{C}{\xi_i^{*k}}\Big|_{\xi_i^{*k}} \quad ,s_i = \frac{D}{\tau_i^k}\Big|_{\tau_i^k} \quad ,s_i^* = \frac{D}{\tau_i^{*k}}\Big|_{\tau_i^{*k}}, \tag{A.40}$$

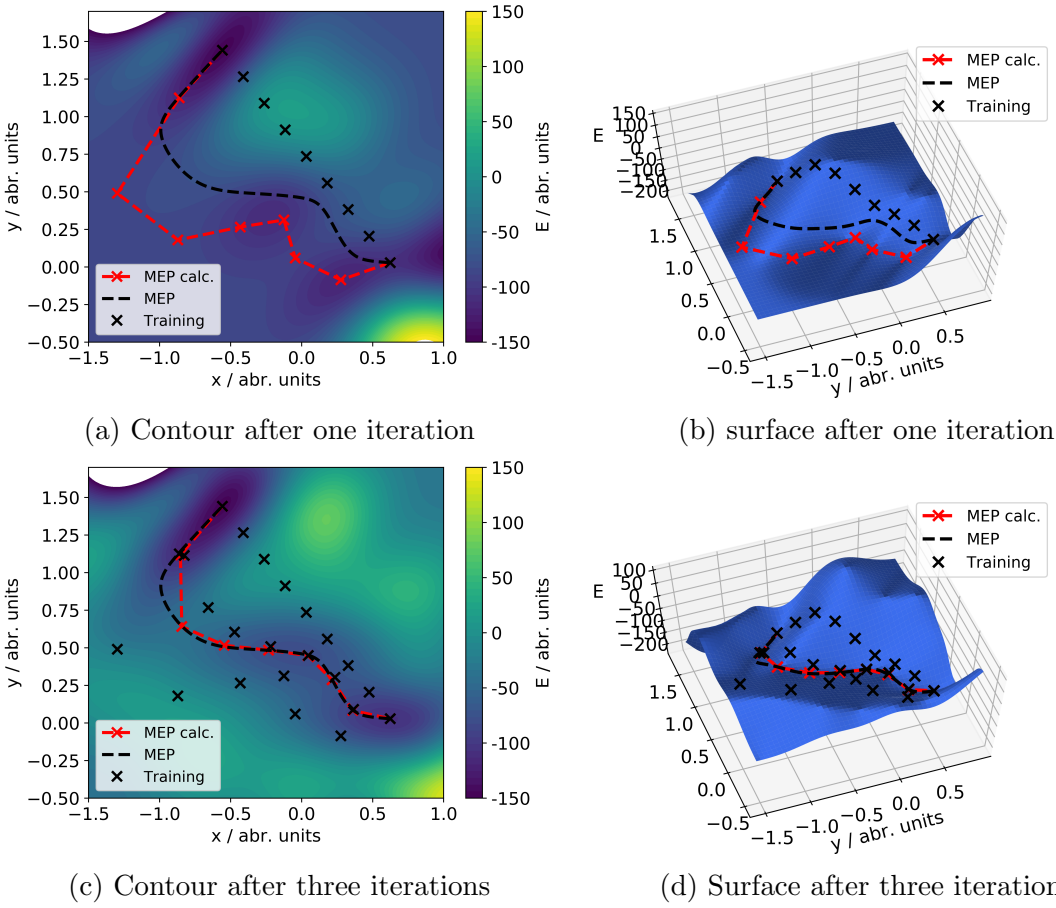
the quadratic approximation of the Lagrangian can be stated as

$$\mathcal{L} = \frac{1}{2}\boldsymbol{w}^\intercal\boldsymbol{w} + \frac{1}{2}\sum_{i=1}^{N} a_i\xi_i^2 + a_i^*\xi_i^{*2} + s_i^*\tau_i^2 + s_i^*\tau_i^{*2} + \text{const.}, \tag{A.41}$$

which is the final approximation.

A.8. Training Example

This example shows the process to approximate the minimum energy path by using a machine learning technique. For that the regularised least squares method is used. The search is done on the Müller-Brown surface. Figure A.1 shows the predicted fictitious energy landscape as well as the obtained minimum energy path after one and after three iteration steps.



(a) Contour after one iteration

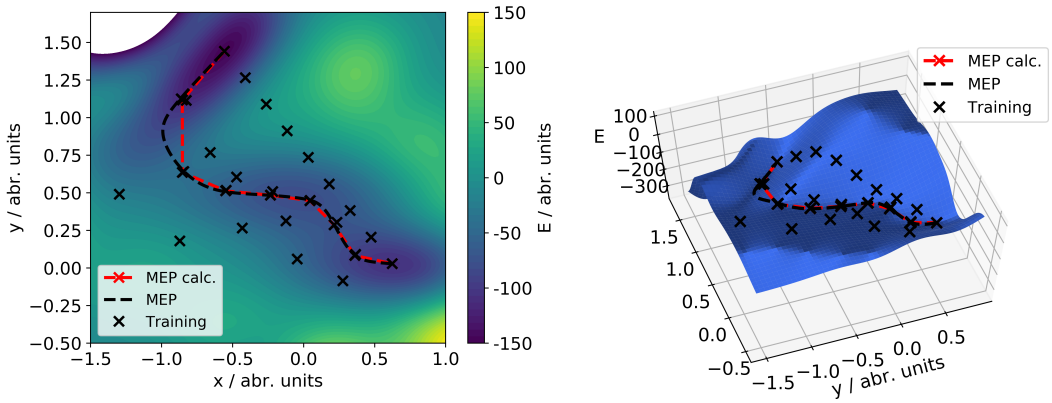
(b) surface after one iteration

(c) Contour after three iterations

(d) Surface after three iterations

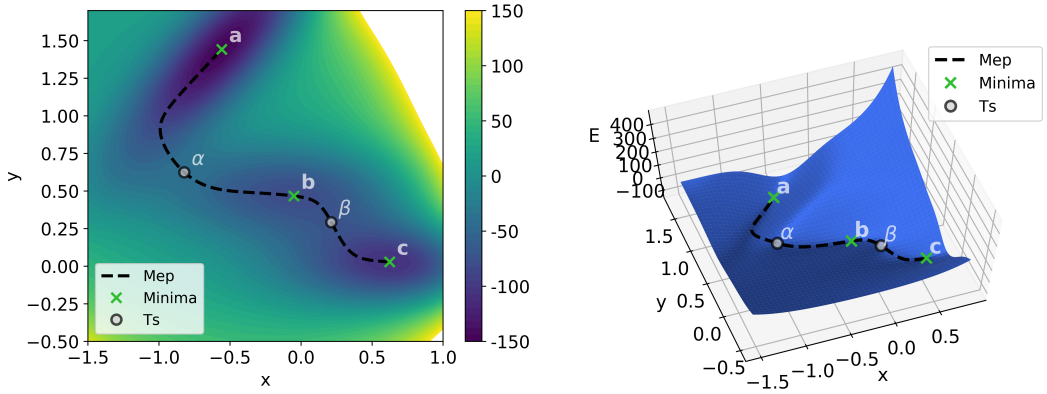
Figure A.1.: Predicted fictitious potential energy surface after different iteration steps.

From Figure A.1a it is obvious to see that the machine learning algorithm is only able to represent the vicinity of the training point reasonably well. After the nudged elastic band converges on the fictitious surface (or even if it exceeds the maximum number of steps) new points are obtained which are then evaluated on the real surface and added to the training set. This is illustrated in Figure A.2. The difference of the two surface in figures A.3 and A.2 nicely denotes the step-wise learning process.



(a) Contour after convergence (b) Surface after convergence

Figure A.2.: Predicted fictitious potential energy surface after convergence of the minimum energy path is achieved.



(a) Contour (b) 3D structure

Figure A.3.: For comparison, the real Müller-Brown surface.

Bibliography

[1] A. A. Peterson, "Acceleration of saddle-point searches with machine learning", *The Journal of Chemical Physics*, vol. 145, no. 7, p. 074 106, 2016. DOI: 10.1063/1.4960708. eprint: https://doi.org/10.1063/1.4960708. [Online]. Available: https://doi.org/10.1063/1.4960708 (cit. on pp. 2, 40).

[2] O.-P. Koistinen, F. B. Dagbjartsdóttir, V. Ásgeirsson, A. Vehtari, and H. Jónsson, "Nuged elastic band calculations accelerated with gaussaian process regression", *The Journal of Chemical Physics*, vol. 147, 2017 (cit. on pp. 2, 16).

[3] A. Denzel and J. Kästner, "Gaussian process regression for geometry optimization", *The Journal of Chemical Physics*, vol. 148, no. 9, p. 094 114, 2018. DOI: 10.1063/1.5017103. eprint: https://doi.org/10.1063/1.5017103. [Online]. Available: https://doi.org/10.1063/1.5017103 (cit. on pp. 2, 16).

[4] A. Szabo and N. S. Ostlund, *Modern Quantum Chemistry: Introduction to Advanced Electronic Structure Theory*, First. Dover Publications, Inc., 1996 (cit. on p. 4).

[5] K. Burke and friends, *The ABC of DFT*. Apr. 10, 2007 (cit. on p. 5).

[6] R. Parr and Y. Weitao, *Density-Functional Theory of Atoms and Molecules*, ser. International Series of Monographs on Chemistry. Oxford University Press, 1994, ISBN: 9780195357738. [Online]. Available: https://books.google.at/books?id=mGOpScSIwU4C (cit. on p. 5).

[7] J. Baker, "An algorithm for the location of transition states", *Journal of Computational Chemistry*, vol. 7, no. 4, pp. 385–395, 1986. DOI: 10.1002/jcc.540070402. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/jcc.540070402. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/jcc.540070402 (cit. on p. 5).

[8] P. M. Zimmerman, "Growing string method with interpolation and optimization in internal coordinates: Method and examples", *The Journal of Chemical Physics*, vol. 138, no. 18, p. 184 102, 2013. DOI: 10.1063/1.4804162. eprint: https://doi.org/10.1063/1.4804162. [Online]. Available: https://doi.org/10.1063/1.4804162 (cit. on p. 5).

[9] A. Behn, P. M. Zimmerman, A. T. Bell, and M. Head-Gordon, "Efficient exploration of reaction paths via a freezing string method", *The Journal of Chemical Physics*, vol. 135, no. 22, p. 224 108, 2011. DOI: 10.1063/1.3664901. eprint: https://doi.org/10.1063/1.3664901. [Online]. Available: https://doi.org/10.1063/1.3664901 (cit. on p. 5).

[10] A. Goodrow, A. T. Bell, and M. Head-Gordon, "Transition state-finding strategies for use with the growing string method", *The Journal of Chemical Physics*, vol. 130, no. 24, p. 244 108, 2009. DOI: 10.1063/1.3156312. eprint: https://doi.org/10.1063/1.3156312. [Online]. Available: https://doi.org/10.1063/1.3156312 (cit. on p. 5).

[11] G. Henkelman and H. Jónsson, "Improved tangent estimate in the nudged elastic band method for finding minimum energy paths and saddle points", *The Journal of Chemical Physics*, vol. 113, no. 22, pp. 9978–9985, 2000. DOI: 10.1063/1.1323224. eprint: https://doi.org/10.1063/1.1323224. [Online]. Available: https://doi.org/10.1063/1.1323224 (cit. on pp. 5, 28, 53).

[12] J. Jónsson Mils, "Nudged elastic band method for finding minimum enery paths of transitions", in *Classical and Quantum Dynamics in Condensed Phase Simulations*, S. D. Schwartz, Ed. World Scientific, 2011, pp. 385–404 (cit. on p. 5).

[13] G. Henkelman, G. Jóhannesson, and H. Jónsson, "Methods for finding saddle points and minimum energy paths", in *Theoretical Methods in Condensed Phase Chemistry*, S. D. Schwartz, Ed. Dordrecht: Springer Netherlands, 2002, pp. 269–302, ISBN: 978-0-306-46949-7. DOI: 10.1007/0-306-46949-9_10. [Online]. Available: https://doi.org/10.1007/0-306-46949-9_10 (cit. on pp. 5, 6).

[14] S. Smidstrup, A. Pedersen, K. Stokbro, and H. Jónsson, "Improved initial guess for minimum energy path calculations", *The Journal of Chemical Physics*, vol. 140, no. 21, p. 214 106, 2014. DOI: 10.1063/1.4878664. eprint: https://doi.org/10.1063/1.4878664. [Online]. Available: https://doi.org/10.1063/1.4878664 (cit. on pp. 6, 54).

[15] R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern classification*, eng, 2. ed. New York ; Weinheim [u.a.]: Wiley, 2001, XX, 654 S. Previous ed.: 1973. - Enth. Teil 1 der 1. Aufl., ISBN: 978-0-471-05669-0 (cit. on p. 9).

[16] (Jul. 15, 2018). Svm: Maximum margin separating hyperplane, [Online]. Available: http://scikit-learn.org/stable/auto_examples/svm/plot_separating_hyperplane.html#sphx-glr-auto-examples-svm-plot-separating-hyperplane-py (cit. on p. 9).

[17] (Jul. 15, 2018). Support vector regression (svr) using linear and non-linear kernels, [Online]. Available: http://scikit-learn.org/stable/auto_examples/svm/plot_svm_regression.html (cit. on p. 9).

[18] (Jul. 30, 2018). Underfitting vs. overfitting, [Online]. Available: http://scikit-learn.org/stable/auto_examples/model_selection/plot_underfitting_overfitting.html (cit. on p. 10).

[19] J. Behler, "Constructing high-dimensional neural network potentials: A tutorial review", *International Journal of Quantum Chemistry*, vol. 115, no. 16, pp. 1032–1050, 2015, ISSN: 1097-461X. DOI: 10.1002/qua.24890. [Online]. Available: http://dx.doi.org/10.1002/qua.24890 (cit. on pp. 11, 33).

[20] A. J. Smola and B. Schölkopf, "A tutorial on support vector regression", *Statistics and Computing*, vol. 14, no. 3, pp. 199–222, Aug. 2004, ISSN: 1573-1375. DOI: 10.1023/B:STCO.0000035301.49549.88. [Online]. Available: https://doi.org/10.1023/B:STCO.0000035301.49549.88 (cit. on pp. 12, 18, 19).

[21] C. E. Rasmussen and C. K. I. Williams, *Gaussian Processes for Machine Learning (Adaptive Computation and Machine Learning)*. The MIT Press, 2005, ISBN: 026218253X (cit. on pp. 12, 14, 16).

[22] "Xvi. functions of positive and negative type, and their connection the theory of integral equations", *Philosophical Transactions of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, vol. 209, no. 441-458, pp. 415–446, 1909, ISSN: 0264-3952. DOI: 10.1098/rsta.1909.0016. eprint: http://rsta.royalsocietypublishing.org/content/209/441-458/415.full.pdf. [Online]. Available: http://rsta.royalsocietypublishing.org/content/209/441-458/415 (cit. on p. 13).

[23] K. T. Jean-Philippe Vert and B. Schölkopf, "A primer on kernel methods", in *in Kernel Methods in Computational*, MIT Press, 2004, pp. 35–70 (cit. on p. 13).

[24] B. Minasny and A. B. McBratney, "The matérn function as a general model for soil variograms", *Geoderma*, vol. 128, no. 3, pp. 192–207, 2005, Pedometrics 2003, ISSN: 0016-7061. DOI: https://doi.org/10.1016/j.geoderma.2005.04.003. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0016706105000911 (cit. on p. 13).

[25] B. Matérn, *Spatial variation*, 2nd, ser. Lecture notes in statistics. Springer-Verlag, 1986, vol. 36, ISBN: 9783540963653 (cit. on p. 13).

[26] M. Lázaro, I. Santamaría, F. Pérez-Cruz, and A. Artés-Rodríguez, "Support vector regression for the simultaneous learning of a multivariate function and its derivatives", *Neurocomputing*, vol. 69, no. 1, pp. 42–61, 2005, Neural Networks in Signal Processing, ISSN: 0925-2312. DOI: https://doi.org/10.1016/j.neucom.2005.02.013. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0925231205001657 (cit. on pp. 13, 19, 22, 56).

[27] Jayadeva, R. Khemchandani, and S. Chandra, "Regularized least squares support vector regression for the simultaneous learning of a function and its derivatives", *Information Sciences*, vol. 178, no. 17, pp. 3402–3414, 2008, ISSN: 0020-0255. DOI: https://doi.org/10.1016/j.ins.2008.04.007. [Online]. Available:

http://www.sciencedirect.com/science/article/pii/S0020025508001291 (cit. on pp. 13, 21).

[28] V. N. Vapnik, *The Nature of Statistical Learning Theory*, ser. Statistics for Engineering and Information Science. Springer, 2000 (cit. on p. 18).

[29] H. W. Kuhn and A. W. Tucker, "Nonlinear programming", in *Proceedings of the Second Berkeley Symposium on Mathematical Statistics and Probability*, Berkeley, Calif.: University of California Press, 1951, pp. 481–492. [Online]. Available: https://projecteuclid.org/euclid.bsmsp/1200500249 (cit. on pp. 19, 56).

[30] J. Suykens and J. Vandewalle, "Least squares support vector machine classifiers", vol. 9, pp. 293–300, Jun. 1999 (cit. on p. 21).

[31] K. Müller and L. D. Brown, "Location of saddle points and minimum energy paths by a constrained simplex optimization procedure", *Theoretica chimica acta*, vol. 53, no. 1, pp. 75–93, Mar. 1979, ISSN: 1432-2234. DOI: 10.1007/BF00547608. [Online]. Available: https://doi.org/10.1007/BF00547608 (cit. on p. 25).

[32] F. Jensen, *Introduction to computational chemistry*. 2007 (cit. on p. 27).

[33] E. Bitzek, P. Koskinen, F. Gähler, M. Moseler, and P. Gumbsch, "Structural relaxation made simple", *Phys. Rev. Lett.*, vol. 97, p. 170 201, 17 Oct. 2006. DOI: 10.1103/PhysRevLett.97.170201. [Online]. Available: https://link.aps.org/doi/10.1103/PhysRevLett.97.170201 (cit. on p. 27).

[34] D. Sheppard, R. Terrell, and G. Henkelman, "Optimization methods for finding minimum energy paths", *The Journal of Chemical Physics*, vol. 128, no. 13, p. 134 106, 2008. DOI: 10.1063/1.2841941. eprint: https://doi.org/10.1063/1.2841941. [Online]. Available: https://doi.org/10.1063/1.2841941 (cit. on p. 27).

[35] L. Verlet, "Computer "experiments" on classical fluids. i. thermodynamical properties of lennard-jones molecules", *Phys. Rev.*, vol. 159, pp. 98–103, 1 Jul. 1967. DOI: 10.1103/PhysRev.159.98. [Online]. Available: https://link.aps.org/doi/10.1103/PhysRev.159.98 (cit. on p. 27).

[36] T. H. Dunning, "Gaussian basis sets for use in correlated molecular calculations. i. the atoms boron through neon and hydrogen", *The Journal of Chemical Physics*, vol. 90, no. 2, pp. 1007–1023, 1989. DOI: 10.1063/1.456153. eprint: https://doi.org/10.1063/1.456153. [Online]. Available: https://doi.org/10.1063/1.456153 (cit. on p. 39).

[37] R. A. Kendall, T. H. Dunning, and R. J. Harrison, "Electron affinities of the first-row atoms revisited. systematic basis sets and wave functions", *The Journal of Chemical Physics*, vol. 96, no. 9, pp. 6796–6806, 1992. DOI: 10.1063/1.462569. eprint: https://doi.org/10.1063/1.462569. [Online]. Available: https://doi.org/10.1063/1.462569 (cit. on p. 39).

[38] W. J. Hehre, R. F. Stewart, and J. A. Pople, "Self-consistent molecular-orbital methods. i. use of gaussian expansions of slater-type atomic orbitals", *The Journal of Chemical Physics*, vol. 51, no. 6, pp. 2657–2664, 1969. DOI: 10.1063/1.1672392. eprint: https://doi.org/10.1063/1.1672392. [Online]. Available: https://doi.org/10.1063/1.1672392 (cit. on p. 39).

[39] Y. Shao, Z. Gan, E. Epifanovsky, A. T. B. Gilbert, M. Wormit, J. Kussmann, A. W. Lange, A. Behn, J. Deng, X. Feng, D. Ghosh, M. Goldey, P. R. Horn, L. D. Jacobson, I. Kaliman, R. Z. Khaliullin, T. Kús, A. Landau, J. Liu, E. I. Proynov, Y. M. Rhee, R. M. Richard, M. A. Rohrdanz, R. P. Steele, E. J. Sundstrom, H. L. Woodcock III, P. M. Zimmerman, D. Zuev, B. Albrecht, E. Alguire, B. Austin, G. J. O. Beran, Y. A. Bernard, E. Berquist, K. Brandhorst, K. B. Bravaya, S. T. Brown, D. Casanova, C.-M. Chang, Y. Chen, S. H. Chien, K. D. Closser, D. L. Crittenden, M. Diedenhofen, R. A. DiStasio Jr., H. Dop, A. D. Dutoi, R. G. Edgar, S. Fatehi, L. Fusti-Molnar, A. Ghysels, A. Golubeva-Zadorozhnaya, J. Gomes, M. W. D. Hanson-Heine, P. H. P. Harbach, A. W. Hauser, E. G. Hohenstein, Z. C. Holden, T.-C. Jagau, H. Ji, B. Kaduk, K. Khistyaev, J. Kim, J. Kim, R. A. King, P. Klunzinger, D. Kosenkov, T. Kowalczyk, C. M. Krauter, K. U. Lao, A. Laurent, K. V. Lawler, S. V. Levchenko, C. Y. Lin, F. Liu, E. Livshits, R. C. Lochan, A. Luenser, P. Manohar, S. F. Manzer, S.-P. Mao, N. Mardirossian, A. V. Marenich, S. A. Maurer, N. J. Mayhall, C. M. Oana, R. Olivares-Amaya, D. P. O'Neill, J. A. Parkhill, T. M. Perrine, R. Peverati, P. A. Pieniazek, A. Prociuk, D. R. Rehn, E. Rosta, N. J. Russ, N. Sergueev, S. M. Sharada, S. Sharmaa, D. W. Small, A. Sodt, T. Stein, D. Stück, Y.-C. Su, A. J. W. Thom, T. Tsuchimochi, L. Vogt, O. Vydrov, T. Wang, M. A. Watson, J. Wenzel, A. White, C. F. Williams, V. Vanovschi, S. Yeganeh, S. R. Yost, Z.-Q. You, I. Y. Zhang, X. Zhang, Y. Zhou, B. R. Brooks, G. K. L. Chan, D. M. Chipman, C. J. Cramer, W. A. Goddard III, M. S. Gordon, W. J. Hehre, A. Klamt, H. F. Schaefer III, M. W. Schmidt, C. D. Sherrill, D. G. Truhlar, A. Warshel, X. Xua, A. Aspuru-Guzik, R. Baer, A. T. Bell, N. A. Besley, J.-D. Chai, A. Dreuw, B. D. Dunietz, T. R. Furlani, S. R. Gwaltney, C.-P. Hsu, Y. Jung, J. Kong, D. S. Lambrecht, W. Liang, C. Ochsenfeld, V. A. Rassolov, L. V. Slipchenko, J. E. Subotnik, T. Van Voorhis, J. M. Herbert, A. I. Krylov, P. M. W. Gill, and M. Head-Gordon, "Advances in molecular quantum chemistry contained in the q-chem 4 program package", *Mol. Phys.*, vol. 113, pp. 184–215, 2015 (cit. on p. 39).

[40] N. Artrith and A. M. Kolpak, "Understanding the composition and activity of electrocatalytic nanoalloys in aqueous solvents: A combination of dft and accurate neural network potentials", *Nano Letters*, vol. 14, no. 5, pp. 2670–2676, 2014, PMID: 24742028. DOI: 10.1021/nl5005674. eprint: https://doi.org/10.1021/nl5005674. [Online]. Available: https://doi.org/10.1021/nl5005674 (cit. on p. 40).

[41] (Nov. 5, 2018). Decomposing signals in components (matrix factorization problems), [Online]. Available: http://scikit-learn.org/stable/modules/decomposition.html#pca (cit. on p. 44).

[42] G. W. Flake, "Support vector machines for regression problems with sequential minimal optimization", *Submitted to NIPS*, vol. 99, 1999 (cit. on p. 47).

[43] J. Platt, "Sequential minimal optimization: A fast algorithm for training support vector machines", p. 21, Apr. 1998. [Online]. Available: https://www.microsoft.com/en-us/research/publication/sequential-minimal-optimization-a-fast-algorithm-for-training-support-vector-machines/ (cit. on p. 47).

[44] A. P. Bartók, R. Kondor, and G. Csányi, "On representing chemical environments", *Phys. Rev. B*, vol. 87, p. 184115, 18 May 2013. DOI: 10.1103/PhysRevB.87.184115. [Online]. Available: https://link.aps.org/doi/10.1103/PhysRevB.87.184115 (cit. on p. 48).

[45] A. Hurley, *Introduction to the electron theory of small molecules*, ser. Essays in chemistry. Academic Press, 1976, ISBN: 9780123624604. [Online]. Available: https://books.google.at/books?id=NcmMAAAAIAAJ (cit. on p. 49).

[46] T. A. Halgren and W. N. Lipscomb, "The synchronous-transit method for determining reaction pathways and locating molecular transition states", *Chemical Physics Letters*, vol. 49, no. 2, pp. 225–232, 1977, ISSN: 0009-2614. DOI: https://doi.org/10.1016/0009-2614(77)80574-5. [Online]. Available: http://www.sciencedirect.com/science/article/pii/0009261477805745 (cit. on p. 54).

[47] R. Fletcher, *Practical Methods of Optimization*, Second. New York, NY, USA: John Wiley & Sons, 2000 (cit. on p. 55).