

Working Title
University of Oslo



Andreas Godø Lefdalsnes

March 25, 2019

Abstract

Abstract goes here [e.g. 1, page 300].

Contents

1	Introduction	4
1.0.1	Molecular dynamics	4
1.0.2	Deep potential molecular dynamics	4
1.0.3	Goals	4
1.0.4	Contributions	4
1.0.5	Structure	4
I	Theory	5
2	Quantum Mechanics	6
2.0.1	Operators	8
2.0.2	Time evolution	9
2.0.3	The Schrodinger equation	10
3	Manybody Quantum Mechanics	12
3.0.1	Hartree-Fock	14
3.0.2	Density functional theory	16
4	Molecular Dynamics	20
4.0.1	From quantum mechanics to molecular dynamics . . .	20
4.0.2	Molecular dynamics simulations	23
4.0.3	Molecular dynamics potentials	23
5	Machine learning	24
5.0.1	Basics of statistical learning	25
5.0.2	Bias-variance decomposition	26
5.0.3	Neural networks	29

5.0.4	Backpropagation	31
5.0.5	Optimization	34
II	Implementation	35
6	Deep potential molecular dynamics	36
7	Quantum Espresso and I-PI	37
8	Pytorch	38
9	LAMMPS	39
III	Results	40
10	Empirical potentials	41
11	Hydrogen systems	42
12	Conclusion	43
	Appendices	45
A	Title 1	46
B	Title 2	47

Chapter 1

Introduction

Intro.

1.0.1 Molecular dynamics

1.0.2 Deep potential molecular dynamics

1.0.3 Goals

1.0.4 Contributions

1.0.5 Structure

Part I

Theory

Chapter 2

Quantum Mechanics

The discussion in this chapter follows closely the discussion in [Sakurai — Modern Quantum Mechanics].

In quantum mechanics, a physical state is represented by a *state vector* in a complex vector space. Such a vector is called a *ket*, denoted by $|\alpha\rangle$. The state ket is postulated to contain all information about the physical state. Two kets can be added to produce a new ket:

$$|\alpha\rangle + |\beta\rangle = |\gamma\rangle.$$

They can also be multiplied by a complex number:

$$c|\alpha\rangle = |\alpha\rangle c = |\delta\rangle.$$

If c is zero the resulting ket is called a *null ket*. If c is non-zero it is postulated that the resulting ket contains the same information.

Observables such as momentum and spin are represented by operators acting on the vector space in question. Operators act on a ket from the left to produce a new ket:

$$A|\alpha\rangle = |\delta\rangle.$$

Of particular importance is when the action of an operator on a ket is the same as multiplication:

$$A|\alpha\rangle = c|\alpha\rangle = |\delta\rangle.$$

These kets are known as *eigenkets* and the corresponding complex numbers are known as *eigenvalues*. The physical state represented by an eigenket is

known as an *eigenstate*. Any ket can be written as an expansion of eigenkets $|a'\rangle$:

$$|\alpha\rangle = \sum_{a'} c_{a'} |a'\rangle,$$

where $c_{a'}$ is a complex coefficient. In principle there are infinitely many linearly independent eigenkets, depending on the dimensionality of the vector space. The uniqueness of the expansion can be proven with orthonogality of the eigenkets, which we will simply postulate.

A *bra space* is a vector space "dual" to the ket space. We postulate that for every ket $|\alpha\rangle$ there exists a bra $\langle\alpha|$. The bra space is spanned by eigenbras $\langle a'|$ corresponding to the eigenkets $|a'\rangle$. The ket and bra spaces have a dual correspondence:

$$\begin{aligned} |\alpha\rangle &\leftrightarrow \langle\alpha| \\ |a'\rangle, |a''\rangle, \dots &\leftrightarrow \langle a'|, \langle a''|, \dots \\ |\alpha\rangle + |\beta\rangle &\leftrightarrow \langle\alpha| + \langle\beta|. \end{aligned}$$

The bra dual to $c|\alpha\rangle$ is postulated to be $c^*|\alpha\rangle$, and more generally:

$$c_\alpha |\alpha\rangle + c_\beta |\beta\rangle \leftrightarrow c_\alpha^* \langle\alpha| + c_\beta^* \langle\beta|.$$

The *inner product* of a bra and a ket is a complex number written as a bra on the left and a ket on the right. It has the fundamental property:

$$\langle\alpha|\beta\rangle = \langle\beta|\alpha\rangle^*,$$

in other words they are complex conjugates. For this to satisfy the requirements of an inner product we must have

$$\langle\alpha|\alpha\rangle \geq 0,$$

with equality if and only if $|\alpha\rangle$ is a null ket. We define the *norm* of a ket as

$$\sqrt{\langle\alpha|\alpha\rangle},$$

which can be used to form normalized kets

$$|\alpha^\sim\rangle = \frac{1}{\sqrt{\langle\alpha|\alpha\rangle}} |\alpha\rangle,$$

with the property

$$\langle\alpha^\sim|\alpha^\sim\rangle = 1.$$

Two kets are said to be *orthogonal* if

$$\langle\alpha|\beta\rangle = 0.$$

2.0.1 Operators

As we briefly mentioned above, operators act on kets from the left to produce a new ket. Two operators A and B are equal

$$A = B$$

if

$$A|\alpha\rangle = B|\alpha\rangle$$

for an arbitrary ket in the relevant ket space. An operator A is said to be the *null operator* if

$$A|\alpha\rangle = 0.$$

Operators can be added, and addition operations are commutative and associative.

$$X + Y = Y + X,$$

$$(X + Y) + Z = X + (Y + Z).$$

Operators act on bras from the right to produce a new bra

$$\langle\alpha|A = \langle\beta|.$$

The ket $A|\alpha\rangle$ and the bra $\langle\alpha|A$ are in general not dual to each other. We define the *hermitian adjoint* A^\dagger as

$$A|\alpha\rangle \leftrightarrow \langle\alpha|A^\dagger.$$

An operator is said to be *hermitian* if

$$A = A^\dagger.$$

Operators can be multiplied. Multiplication is associative, but non-commutative:

$$XY \neq YX,$$

$$X(YZ) = (XY)Z.$$

The left product of a ket and a bra is known as the *outer product*:

$$|\alpha\rangle\langle\beta|.$$

The outer product should be treated as an operator, while the inner product $\langle\alpha|\beta\rangle$ is a complex number.

If an operator is to the left of a ket $|\alpha\rangle A$ or to the right of a bra $A\langle\beta|$ these are illegal products, in other words not defined within the ruleset of quantum mechanics. The associative properties of operators are postulated to hold true as long as we are dealing with legal multiplications among kets, bras and operators. As an example, the outer product acting on a ket:

$$(|\alpha\rangle\langle\beta|)|\gamma\rangle,$$

can be equivalently regarded as scalar multiplication

$$|\alpha\rangle(\langle\alpha|\gamma\rangle) = |\alpha\rangle c = c|\alpha\rangle,$$

where $c = \langle\alpha|\gamma\rangle$ is just a complex number.

2.0.2 Time evolution

In quantum mechanics, time is treated not as an observable, but as a parameter. Relativistic quantum mechanics treats space and time on the same footing, but only by demoting position to a parameter.

Suppose we have a physical system $|\alpha\rangle$ at a time t_0 . Denote the ket at a later time $t > t_0$ by

$$|\alpha, t; t_0\rangle.$$

As time is assumed to be a continuous parameter we expect as we evolve the system backward in time

$$\lim_{t \rightarrow t_0} |\alpha, t; t_0\rangle = |\alpha\rangle.$$

The kets separated by a time $t - t_0$ are related by the *time-evolution operator* \mathcal{U} :

$$|\alpha, t; t_0\rangle = \mathcal{U}(t, t_0) |\alpha, t_0\rangle.$$

If the state ket is normalized to unity at a time t_0 , it must remain normalized at a later time:

$$\langle\alpha, t_0|\alpha, t_0\rangle = \langle\alpha, t; t_0|\alpha, t; t_0\rangle = 1.$$

This is guaranteed if the time evolution operator \mathcal{U} is a *unitary* operator:

$$\mathcal{U}^\dagger \mathcal{U} = 1.$$

We also require the composition property:

$$\mathcal{U}(t_2, t_0) = \mathcal{U}(t_2, t_1)\mathcal{U}(t_1, t_0), \quad (t_2 > t_1 > t_0).$$

If we consider an infinitesimal time-evolution operator

$$|\alpha, t_0 + dt; t_0\rangle = \mathcal{U}(t_0 + dt, t_0) |\alpha, t_0\rangle,$$

it must reduce to the identity operator as the infinitesimal time interval dt goes to zero:

$$\lim_{dt \rightarrow 0} \mathcal{U}(t_0 + dt, t_0) = 1,$$

and we expect the difference between the operators to be of first order in dt . These requirements are all satisfied by

$$\mathcal{U}(t_0 + dt, t_0) = 1 - i\Omega dt,$$

where Ω is a Hermitian operator:

$$\Omega^\dagger = \Omega.$$

The operator Ω has the dimension inverse time. Frequency or inverse time is related to energy through the Planck-Einstein relation:

$$E = \hbar\omega.$$

In classical mechanics the Hamiltonian is the generator of time evolution, so we postulate that Ω is related to the Hamiltonian operator H :

$$\Omega = \frac{H}{\hbar}.$$

The Hamiltonian operator represents the energy of our system, which is a physical observable and must therefore be Hermitian.

2.0.3 The Schrodinger equation

By exploiting the composition property of the time-evolution operator we obtain:

$$\mathcal{U}(t + dt, t_0) = \mathcal{U}(t + dt, t)\mathcal{U}(t, t_0) = \left(1 - \frac{iHdt}{\hbar}\right)\mathcal{U}(t, t_0),$$

where the time difference $t - t_0$ is not required to be infinitesimal. This implies

$$\mathcal{U}(t + dt, t_0) - \mathcal{U}(t, t_0) = -i\left(\frac{H}{\hbar}\right)dt\mathcal{U}(t, t_0),$$

and taking the limit $dt \rightarrow 0$:

$$i\hbar\frac{\partial}{\partial t}\mathcal{U}(t, t_0) = H\mathcal{U}(t, t_0).$$

This is known as the Schrodinger equation for the time-evolution operator. Multiplying both sides by a ket $|\alpha, t_0\rangle$ leads to the Schrodinger equation:

$$i\hbar\frac{\partial}{\partial t}\mathcal{U}(t, t_0) |\alpha, t_0\rangle = H\mathcal{U}(t, t_0) |\alpha, t_0\rangle .$$

As the ket does not depend on time this is the same as

$$i\hbar\frac{\partial}{\partial t} |\alpha, t_0\rangle = H |\alpha, t_0\rangle .$$

Chapter 3

Manybody Quantum Mechanics

This chapter will give a brief overview of the Hartree-Fock and Density Functional Theory methods, which are the primary workhorses in manybody quantum theory.

In most applications of quantum theory we are interested in finding solutions to the non-relativistic time-independent Schrodinger equation:

$$\hat{H} |\Psi\rangle = E |\Psi\rangle ,$$

with the Hamiltonian \hat{H} describing a system of nuclei and electrons with coordinates \mathbf{R}_i , $i = 1, 2, \dots, A$ and \mathbf{r}_i , $i = 1, 2, \dots, N$ respectively. The distance between nuclei a and electron i is given as $r_{ai} = |\mathbf{R}_a - \mathbf{r}_i|$ and correspondingly for the nuclei-nuclei and electron-electron distances.

The full Hamiltonian for a set of N electrons and A nuclei in atomic units is

$$\begin{aligned} \hat{H} = & - \sum_{i=1}^N \frac{1}{2} \nabla_i^2 - \sum_{a=1}^A \frac{1}{M_a} \nabla_a^2 - \sum_{i=1}^N \sum_{a=1}^A \frac{Z_a}{r_{ia}} \\ & + \sum_{i=1}^N \sum_{j=i+1}^N \frac{1}{r_{ij}} + \sum_{a=1}^A \sum_{b=a+1}^A \frac{Z_a Z_b}{R_{ab}} . \end{aligned} \quad (3.1)$$

The first two terms describe the kinetic energy operators of the electrons and nuclei, with M_a the ratio of the mass of nuclei a to the electron mass. The third term describes the coulomb attraction between electron and nuclei,

while the fourth and fifth terms describe the repulsion between electrons and nuclei respectively.

From this description we are most often interested in the electronic structure problem presented by applying the *Born-Oppenheimer* approximation. Since the nuclei are approximately 2000 times heavier than the electrons, the electrons can to a good approximation be described as moving in the field of fixed nuclei. This means we can neglect the kinetic energy terms of the nuclei, while considering an averaged effect from the nuclei-nuclei repulsion. The nuclei-nuclei repulsion energy adds a constant to the energy eigenvalues, but has no effect on the energy eigenfunctions. The remaining terms are known as the electronic Hamiltonian:

$$\hat{H}_e = - \sum_{i=1}^N \frac{1}{2} \nabla_i^2 - \sum_{i=1}^N \sum_{a=1}^A \frac{Z_A}{r_{ia}} + \sum_{i=1}^N \sum_{j=i+1}^N \frac{1}{r_{ij}}. \quad (3.2)$$

The electronic wavefunction $\Psi_e = \Psi_e(\{r_i\}; \{R_a\})$ is a function of the electronic coordinates with a parametric dependence on the fixed nucleic coordinates. The electronic energy is obtained in the usual way $E_e = \langle \Psi_e | \hat{H}_e | \Psi_e \rangle$. The total energy of our system must now include the constant nuclear repulsion:

$$E_{tot} = E_e + \sum_{a=1}^A \sum_{b=a+1}^A \frac{Z_a Z_b}{R_{ab}}.$$

If one has solved the Schrodinger equation for the electronic Hamiltonian, one can subsequently solve for the nucleic motion using the same method, i.e. substituting the electronic coordinates for their average values, averaged over the electronic wave function. We are then left with a nuclear Hamiltonian \hat{H}_n :

$$\begin{aligned} \hat{H}_n &= - \sum_{a=1}^A \frac{1}{2M_a} \nabla_a^2 + \left\langle - \sum_{i=1}^N \frac{1}{2} \nabla_i^2 - \sum_{i=1}^N \sum_{j=i+1}^N \frac{1}{r_{ij}} \right\rangle \\ &\quad + \sum_{a=1}^A \sum_{b=a+1}^A \frac{Z_a Z_b}{R_{ab}} \\ &= - \sum_{a=1}^A \frac{1}{2M_a} \nabla_a^2 + E_{tot}. \end{aligned} \quad (3.3)$$

Under this approximation the nuclei move on a potential energy surface obtained by solving the electronic Hamiltonian.

Finally, we would also like to go one step further and neglect the relative positions of the nuclei. To this end we consider only the center of mass of the nucleus with a total charge Z , which gives us the final expression we want for our Hamiltonian:

$$\hat{H} = -\sum_{i=1}^N \frac{1}{2} \nabla_i^2 - \sum_{i=1}^N \frac{Z}{r_i} + \sum_{i=1}^N \sum_{j=i+1}^N \frac{1}{r_{ij}} \quad (3.4)$$

3.0.1 Hartree-Fock

[cite](<http://vergil.chemistry.gatech.edu/notes/hf-intro/node3.html>) The Hartree-Fock method is a method for finding solutions to the electronic Hamiltonian assuming the electron-electron repulsion can be approximated with a set of single-particle functions or *orbitals* moving in a mean field generated by the presence of other electrons. Assuming that the electrons do not interact the Hamiltonian is separable and the wavefunction is simply a product of orbitals ψ which are solutions to a onebody Hamiltonian. This gives us an ansatz for the manybody wavefunction Ψ known as the *Hartree product*:

$$\Psi(\mathbf{r}_1, \dots, \mathbf{r}_N) = \psi(\mathbf{r}_1) \cdots \psi(\mathbf{r}_N).$$

Since we are dealing with fermions this ansatz fails to satisfy the antisymmetry principle, i.e. the wavefunction is not antisymmetric with respect to the interchange of any two particles. Fermions in addition to three spatial degrees of freedom also have a spin degree of freedom σ which means the fermion can be described by the space-spin coordinate $\mathbf{x} = (\mathbf{r}, \sigma)$ with $\mathbf{x} \in \mathbb{R}^3 \otimes \sigma$.

The problem of antisymmetry in a system of N fermions is satisfied by the introduction of *Slater determinants*

$$\Psi(\mathbf{x}_1, \dots, \mathbf{x}_N) = \frac{1}{\sqrt{N!}} \begin{vmatrix} \chi_1(\mathbf{x}_1) & \chi_2(\mathbf{x}_1) & \dots & \chi_N(\mathbf{x}_1) \\ \chi_1(\mathbf{x}_2) & \chi_2(\mathbf{x}_2) & \dots & \chi_N(\mathbf{x}_2) \\ \dots & \dots & \dots & \dots \\ \chi_1(\mathbf{x}_N) & \chi_2(\mathbf{x}_N) & \dots & \chi_N(\mathbf{x}_N) \end{vmatrix}, \quad (3.5)$$

with $\chi(\mathbf{x})$ spin orbitals and a normalization factor $(N!)^{-1/2}$. The introduction of this ansatz is equivalent to assuming that all electrons move

independently of each other in a mean field generated by the electron-electron repulsion.

Define the one-electron operator of the electronic Hamiltonian as

$$\hat{h}_1(\mathbf{x}_i) = -\frac{1}{2}\nabla_i^2 - \frac{Z}{r_i},$$

with a twobody interaction term

$$\hat{v}(\mathbf{x}_i, \mathbf{x}_j) = \frac{1}{r_{ij}},$$

with the electronic Hamiltonian written more compactly as

$$\hat{H} = \sum_i \hat{h}_1(\mathbf{x}_i) + \sum_{i<j} \hat{v}(\mathbf{x}_i, \mathbf{x}_j).$$

The expectation value of the energy is given as

$$\langle \Psi | \hat{H} | \Psi \rangle.$$

The *variational theorem* says the expectation value of any normalized wavefunction with respect to the energy is an upper bound to the ground state energy. This suggests a procedure wherein we vary the parameters of a set of approximate wavefunction Ψ_T until an energy minimum is reached.

The Hartree-Fock energy can be written in terms of integrals over the onebody and interaction terms:

$$E_{HF} = \sum_i \langle i | \hat{h} | i \rangle + \sum_{i<j} \langle ij | \hat{v} | ij \rangle_{AS},$$

where we have introduced an antisymmetrized matrix element

$$\langle ij | \hat{v} | ij \rangle_{AS} = \langle ij | \hat{v} | ij \rangle - \langle ij | \hat{v} | ji \rangle,$$

and the shorthand integrals

$$\langle i | \hat{h}_1 | i \rangle = \int d\mathbf{r} \chi_i^* \hat{h}_1 \chi_i,$$

and

$$\langle ij | \hat{v} | ij \rangle = \int d\mathbf{r}_i d\mathbf{r}_j \chi_i^* \chi_j^* \hat{v} \chi_i \chi_j.$$

In order to solve these integrals numerically we perform a linear expansion of the spin orbitals χ in terms of a fixed orthogonal basis ϕ :

$$\chi_i = \sum_{\lambda} C_{i\lambda} \phi_{\lambda},$$

in principle an infinite sum, but in practice truncated. This allows us to rewrite the Hartree-Fock energy as

$$E_{HF} = \sum_i \sum_{\alpha, \beta} C_{i\alpha}^* C_{i\beta} \langle \alpha | \hat{h}_1 | \beta \rangle + \sum_{i < j} \sum_{\alpha, \beta, \delta, \eta} C_{i\alpha}^* C_{j\beta}^* C_{i\delta} C_{j\eta} \langle \alpha \beta | \hat{v} | \delta \eta \rangle.$$

Work out the remainder here...

3.0.2 Density functional theory

Density functional theory is a method for investigating the electronic structure of a manybody system by finding approximations to the ground state density $\rho(\mathbf{r})$. Our starting point is again the electronic Hamiltonian

$$\hat{H} = - \sum_{i=1}^N \frac{1}{2} \nabla_i^2 - \sum_{i=1}^N \frac{Z}{r_i} + \sum_{i=1}^N \sum_{j=i+1}^N \frac{1}{r_{ij}}. \quad (3.6)$$

Any electronic wavefunction Ψ which solves this equation is in principle a function of $4N$ coordinates $\mathbf{x}_i = (\mathbf{r}_i, \sigma_i)$, $i = 1, \dots, N$ where N is the number of electrons. Once we have obtained a solution to the Schrodinger equation we can obtain the one-electron density $\rho(\mathbf{r})$ as:

$$\rho(\mathbf{r}) = \int |\Psi(\mathbf{r}, \sigma, \mathbf{x}_2, \dots, \mathbf{x}_N)|^2 d\sigma d\mathbf{x}_2 \dots d\mathbf{x}_N. \quad (3.7)$$

Since the wavefunction is a unique functional of the Hamiltonian \hat{H} , the one-electron density is uniquely determined by the Hamiltonian. Hohenberg and Kohn showed in 1964 in their first theorem that this mapping can be inverted, i.e. that the one-electron density uniquely determines the Hamiltonian of our system (up to an arbitrary constant). Taken altogether, this means that all properties of our system, including the Hamiltonian and the manybody wavefunction are fixed by a one-electron density carrying a dependency on only 3 spatial coordinates.

The electronic Hamiltonian can be rewritten as

$$\hat{H} = \hat{F} + \hat{V}_{ne},$$

where \hat{F} is an operator consisting of the kinetic energy and electron-electron operators and \hat{V}_{ne} is the electron-nuclei interaction.

For their second theorem, Hohenberg and Kohn defined the universal density functional

$$F[n] = \langle \Psi[n] | \hat{F} | \Psi[n] \rangle ,$$

and the total electronic energy functional

$$E[n] = F[n] + \int \hat{V}_{ne} n(\mathbf{r}) d\mathbf{r}.$$

Hohenberg and Kohn showed the energy functional with respect to N -electron densities $n(\mathbf{r})$ is an upper bound to the ground state energy:

$$E_0 \leq F[n] + \int \hat{V}_{ne} n(\mathbf{r}) d\mathbf{r},$$

with equality if and only if the one-electron density is the one-electron density corresponding to our system \hat{H} . This suggests a variational procedure, wherein we minimize the total electronic energy functional until we reach an energy minimum:

$$E_{min} = \min_n E[n],$$

which serves as our best estimate for the ground state one-electron density $\rho_0(\mathbf{r})$.

Levy and Lieb proposed to redefine the universal density functional in terms of normalized antisymmetric wavefunctions Ψ which yield a fixed density ρ :

$$F[n] = \min_{\Psi} \rho \langle \Psi | \hat{F} | \Psi \rangle = \langle \Psi[n] | \hat{F} | \Psi[n] \rangle ,$$

wherein the minima search is performed over wavefunctions which yield the fixed density ρ . A search is then performed over densities ρ until we reach an energy minimum. This method is known as the *constrained search formulation*. However, we still do not have an explicit expression for the

universal density functional $F[n]$, and direct approximations have proved difficult.

Kohn and Sham proposed to decompose $F[n]$ as

$$F[n] = T_s[n] + E_{Hxc}[n],$$

where $T_s[n]$ is a non-interacting kinetic-energy functional which can be defined through the constrained-search formulation:

$$T_s[n] = \min_{\Phi \rightarrow n} \langle \Phi | \hat{T} | \Phi \rangle = \langle \Phi[n] | \hat{T} | \Phi[n] \rangle,$$

wherein the minima search is now performed over normalized single-determinant wavefunctions Φ which yield the fixed density ρ . The functional $E_{Hxc}[n]$ is known as the Hartree-exchange-correlation functional. The variational procedure is then performed over single-determinant wavefunctions which yield a fixed density ρ and then minimized over densities:

$$\begin{aligned} E_0 &= \min_n \{ F[n] + \int \hat{V}_{ne} n(\mathbf{r}) d\mathbf{r} \} \\ &= \min_n \min_{\Phi \rightarrow n} \{ \langle \Phi | \hat{T} + \hat{V}_{ne} | \Phi \rangle + E_{Hxc}[n_\Phi] \} \\ &= \min_{\Phi} \{ \langle \Psi | \hat{T} + \hat{V}_{ne} | \Phi \rangle + E_{Hxc}[n_\Phi] \}. \end{aligned} \quad (3.8)$$

As with the Hartree-Fock method, the single determinant wavefunctions are constructed from an orthonormal basis of spin orbitals $\chi_i(\mathbf{x})$, $i = 1, \dots, N$ with $\mathbf{x} = (\mathbf{r}, \sigma)$. The total electronic energy can be expressed in terms of spatial orbitals $\phi_i(\mathbf{r})$ after integrating over the spin variables:

$$E = \sum_i \int \phi_i^*(\mathbf{r}) \left(-\frac{1}{2} \nabla^2 + \hat{v}_{ne} \right) \phi_i(\mathbf{r}) d\mathbf{r} + E_{Hxc}[n], \quad (3.9)$$

with the density expressed as

$$n(\mathbf{r}) = \sum_i |\phi_i(\mathbf{r})|^2. \quad (3.10)$$

With the constraint that the spatial orbitals be normalized we obtain the following Lagrangian:

$$\mathcal{L}[\{\phi_i\}] = E[\{\phi_i\}] - \sum_i \epsilon_i \left(\int \phi_i^*(\mathbf{r}) \phi_i(\mathbf{r}) d\mathbf{r} - 1 \right),$$

with ϵ_i the associated Lagrangian multiplier. We find the energy minimum where the Lagrangian is stationary:

$$\frac{\partial \mathcal{L}}{\partial \phi_i^*(\mathbf{r})} = 0.$$

The spatial orbitals are expanded as a linear combination of a known basis, such as hydrogen-like functions or Gaussian-type orbitals:

$$\phi_i(\mathbf{r}) = \sum_{\lambda} C_{\lambda i} \chi_i(\mathbf{r}).$$

Chapter 4

Molecular Dynamics

Our starting point for molecular dynamics is the full Hamiltonian for a set of N electrons and A nuclei:

$$\begin{aligned} \hat{H} = & - \sum_{i=1}^N \frac{1}{2} \nabla_i^2 - \sum_{a=1}^A \frac{1}{M_a} \nabla_a^2 - \sum_{i=1}^N \sum_{a=1}^A \frac{Z_a}{r_{ia}} \\ & + \sum_{i=1}^N \sum_{j=i+1}^N \frac{1}{r_{ij}} + \sum_{a=1}^A \sum_{b=a+1}^A \frac{Z_a Z_b}{R_{ab}} \end{aligned} \quad (4.1)$$

We want to find solutions to the time-dependent non-relativistic Schrodinger equation:

$$i\hbar \frac{\partial}{\partial t} \Psi = \hat{H} \Psi.$$

4.0.1 From quantum mechanics to molecular dynamics

We will follow the route of Tully described in [<https://core.ac.uk/download/pdf/35009882.pdf?repos>]. The wavefunction is separated in terms of the electronic and nuclear coordinates with an ansatz

$$\Psi(\{\mathbf{r}_i\}, \{\mathbf{R}_I\}, t) \approx \Psi(\{\mathbf{r}_i\}) \chi(\{\mathbf{R}_I\}) \exp \left[\frac{i}{\hbar} \int_{t_0}^t dt' \hat{E}_e(t') \right],$$

with the electronic and nuclear wavefunctions normalized to unity at every instance of time. A phase factor is introduced to make the equations look nice:

$$\hat{E}_e = \int d\mathbf{r} d\mathbf{R} \Psi * \chi * \hat{H} \Psi \chi,$$

where the integration occurs over all spatial coordinates $\{\mathbf{r}_i\}, \{\mathbf{R}_I\}$. This is a single determinant ansatz which must lead to a mean-field description of the dynamics. This ansatz differs from the Born-Oppenheimer approximation:

$$\Psi_{BO} = \sum_{k=0}^{\infty} \Psi_k \chi_k,$$

even in the single-determinant limit where only a single state k is included in the expansion.

Inserting this ansatz into the Schrodinger equation reveals the following set of equations:

$$\begin{aligned} i\hbar \frac{\partial \Psi}{\partial t} &= - \sum_i \frac{\hbar^2}{2m_e} \nabla_i^2 \Psi + \left\{ \int d\mathbf{R} \chi^* V_{ne} \chi \right\} \Psi, \\ i\hbar \frac{\partial \chi}{\partial t} &= - \sum_I \frac{\hbar^2}{2M_I} \nabla_I^2 \chi + \left\{ \int d\mathbf{r} \Psi^* \hat{H} \Psi \right\} \chi. \end{aligned}$$

This set of equations is the basis for the time-dependent self-consistent field (TDSCF) method, wherein particles move in time-dependent effective potentials obtained from quantum mechanical expectation values.

In the framework of classical molecular dynamics we approximate the nuclei as classical point particles. This can be done by rewriting the nuclear wavefunction as

$$\chi = A \exp[iS/\hbar],$$

with an amplitude factor A and a phase S which are both considered to be real. The TDSCF equations are rewritten in terms of these variables

$$\frac{\partial S}{\partial t} + \sum_I \frac{1}{2M_I} (\nabla_I S)^2 + \int d\mathbf{r} \Psi^* \hat{H} \Psi = \hbar^2 \sum_I \frac{1}{2M_I} \frac{\nabla_I^2 A}{A},$$

$$\frac{\partial A}{\partial t} + \sum_I \frac{1}{M_I} (\nabla_I A) (\nabla_I S) + \sum_I \frac{1}{2M_I} A (\nabla_I^2 S) = 0.$$

This set of equations is known as the "quantum fluid dynamical representation". The term for S contains a term for \hbar which vanishes in the classical limit $\hbar \rightarrow 0$:

$$\frac{\partial S}{\partial t} + \sum_I \frac{1}{2M_I} (\nabla_I S)^2 + \int d\mathbf{r} \Psi^* \hat{H} \Psi = 0.$$

This formulation of the nuclear dynamics is isomorphic to the Hamilton-Jacobi formulation:

$$\frac{\partial S}{\partial t} + \hat{H} = 0,$$

with the classical Hamilton function

$$\hat{H} = T(P_I) + V(R_I),$$

with coordinates R_I and conjugate momenta P_I .

If we identify the conjugate momenta

$$\mathbf{P}_I = \nabla_I S,$$

we obtain the Newtonian equations of motion:

$$\begin{aligned} \frac{d\mathbf{P}}{dt} &= -\nabla_I V = -\nabla_I \int d\mathbf{r} \Psi^* \hat{H} \Psi \quad \text{or} \\ M_I \frac{d^2 \mathbf{R}_I}{dt^2} &= -\nabla_I \int d\mathbf{r} \Psi^* \hat{H} \Psi \\ &= -\nabla_I V_e^E(R_I(t)). \end{aligned} \tag{4.2}$$

The nuclei now move according to classical mechanics in an effective potential V_e^E generated by the electrons. This potential is a function only of the nuclear degrees of freedom at time t after averaging out the electronic degrees of freedom.

For consistency the nuclear wavefunction appearing in the TDSCF equation for the electronic degrees of freedom has to be replaced by the positions of the nuclei point particles. This is done by replacing the nuclear density

$|\chi|^2$ in the limit $\hbar \rightarrow 0$ by a product of delta functions $\prod_I \delta(\mathbf{R}_I - \mathbf{R}_I(t))$ centered at the instantaneous positions $\{\mathbf{R}_I(t)\}$ of the classical nuclei. This leads to a time-dependent wave equation for the electrons:

$$i\hbar \frac{\partial \Psi}{\partial t} = - \sum_i \frac{\hbar}{2m_e} \nabla_i^2 \Psi + V_{ne} \Psi,$$

which evolve quantum mechanically as the nuclei propagate classically. This mixed approach is commonly referred to as *Ehrenfest molecular dynamics*. Although the underlying equations describe a mean-field theory, the Ehrenfest approach includes transitions between electronic states.

4.0.2 Molecular dynamics simulations

4.0.3 Molecular dynamics potentials

Chapter 5

Machine learning

Machine learning is the study of algorithms and statistical models employed by computing systems capable of performing tasks without explicit instruction. While traditional algorithms rely on some specified input and a ruleset for determining the output, machine learning is instead concerned with a set of generic algorithms which can find patterns in a broad class of data sets. This section will give a brief overview of machine learning, and more specifically the class of algorithms known as neural networks, and will follow closely the review by [Mehta et. al.] which the reader is encouraged to seek out.

Examples of machine learning problems include identifying objects in images, transcribing text from audio and making film recommendations to viewers based on their watch history. Machine learning problems are often subdivided into estimation and prediction problems. In both cases, we choose some observable \mathbf{x} (e.g. the period of a pendulum) related to some parameters $\boldsymbol{\theta}$ (e.g. the length and the gravitational constant) through a model $p(\mathbf{x}|\boldsymbol{\theta})$ that describes the probability of observing \mathbf{x} given $\boldsymbol{\theta}$. Subsequently we perform an experiment to obtain a dataset \mathbf{X} and use these data to fit the model. Fitting the model means finding the parameters $\hat{\boldsymbol{\theta}}$ that provide the best explanation for the data. *Estimation* problems are concerned with the accuracy of $\hat{\boldsymbol{\theta}}$, whereas prediction problems are concerned with the ability of the model $p(\mathbf{x}|\boldsymbol{\theta})$ to make new predictions. Physics has traditionally been more concerned with the estimation of model parameters, while in this thesis we will be focused on the accuracy of the model.

Many problems in machine learning are defined by the same set of ingredients. The first is the dataset $\mathcal{D} = (\mathbf{X}, \mathbf{Y})$, where \mathbf{X} is a matrix containing

observations of the independent variables \mathbf{x} , and \mathbf{Y} is a matrix containing observations of dependent variables. Second is a model $\mathbf{F} : \mathbf{x} \rightarrow \mathbf{y}$ which is a function of the parameters $\boldsymbol{\theta}$. Finally we have a cost function $\mathcal{C}(\mathbf{Y}, \mathbf{F}(\mathbf{X}; \boldsymbol{\theta}))$ that judges the performance of our model at generating predictions.

In the case of linear regression we consider a set of independent observations $\mathbf{X} = [\mathbf{x}_1 \ \mathbf{x}_2 \ \dots \ \mathbf{x}_N]$ related to a set of dependent observations $\mathbf{y} = (y_1, y_2, \dots, y_N)$ through a linear model $f(\mathbf{x}; \boldsymbol{\theta}) = x_1 \cdot w_1 + x_2 \cdot w_2 + \dots + x_P \cdot w_P$, with parameters $\boldsymbol{\theta} = (w_1, w_2, \dots, w_P)$. The cost function is the well known sum of least squares $\mathcal{C}(\mathbf{y}, f(\mathbf{X}; \boldsymbol{\theta})) = \sum_i^N (y_i - f(\mathbf{x}_i; \boldsymbol{\theta}))^2$ and the best fit is chosen as the set of parameters which minimize this cost function: $\hat{\boldsymbol{\theta}} = \underset{\boldsymbol{\theta}}{\operatorname{argmin}} \mathcal{C}(\mathbf{Y}, f(\mathbf{X}; \boldsymbol{\theta}))$.

5.0.1 Basics of statistical learning

Statistical learning theory is a field of statistics dealing with the problem of making predictions from data. We begin with an unknown function $y = f(x)$ and our goal is to develop a function $h(x)$ such that $h \sim f$. We fix a hypothesis set \mathcal{H} that the algorithm is willing to consider. The expected error of a particular h over all possible inputs x and outputs y is:

$$E[h] = \int_{X \times Y} \mathcal{C}(h(x), y) \rho(x, y) dx dy,$$

where \mathcal{C} is a cost function and $\rho(x, y)$ is the joint probability distribution for x and y . This is known as the *expected error*. Since this is impossible to compute without knowledge of the probability distribution ρ , we instead turn to the *empirical error*. Given n data points the empirical error is given as:

$$E_S[h] = \frac{1}{n} \sum_i^n \mathcal{C}(h(x_i), y_i).$$

The *generalization error* is defined as the difference between the expected and empirical errors:

$$G = E[h] - E_S[h].$$

We say an algorithm is able to learn from data or *generalize* if

$$\lim_{n \rightarrow \infty} G = 0.$$

We are in general unable to compute the expected error, and therefore unable to compute the generalization error. The most common approach known as *cross-validation* is to estimate the generalization error by subdividing our dataset into a *training* set and a *test* set. The value of the cost function on the training set is called the *in-sample* error and the value of the cost function on the test set the *out-of-sample* error. Assuming the dataset is sufficiently large and representative of f , and the subsampling into train and test datasets is unbiased, the in-sample error can serve as an appropriate proxy for the generalization error.

In figure 5.1 we show the typical evolution of the errors as the number of data points increase. It is assumed that the function being learned is sufficiently complicated that we cannot learn it exactly, and that we have a sizeable number of data points available. The in-sample error will decrease monotonically, as our model is not able to learn the underlying data exactly. In contrast, the out-of-sample error will decrease, as the sampling noise decreases and the training data set becomes more representative of the underlying probability distribution. In the limit, these errors both approach same value, which is known the model *bias*. The bias represents the best our model could do in the infinite data limit. The out-of-sample error produced from the sampling noise is known as *variance*, and will vanish completely given an infinite representative data set.

In figure 5.2 we show the typical evolution of the out-of-sample error as the model *complexity* increases. Model complexity is a measure of the degrees of freedom in the model space, for example the number of coefficients in a polynomial regression. In the figure we can see that bias decreases monotonically as model complexity increases, as the model is able to fit a larger space of functions. However, the variance will also increase as the model becomes more susceptible to sampling noise. In general the lowest out-of-sample error, and therefore generalization error, is achieved at an intermediate model complexity. We also find that as model complexity increases, a larger amount of data points is required to be able to reasonably fit the true function.

5.0.2 Bias-variance decomposition

Consider a dataset $\mathcal{D}(\mathbf{X}, \mathbf{y})$ of n pairs of independent and dependent variables. Assume the true data is generated from a noisy model:

$$y = f(\mathbf{x}) + \epsilon,$$

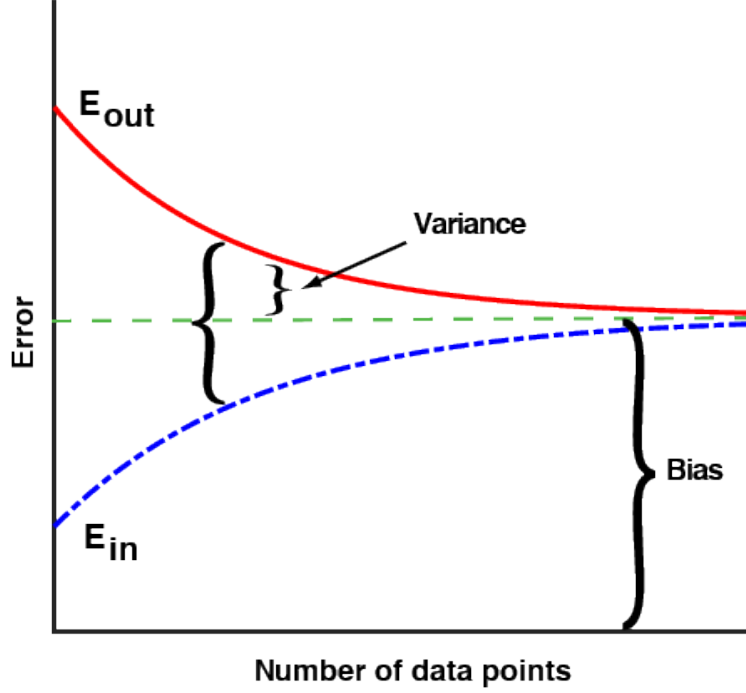


Figure 5.1: Typical in-sample and out-of-sample error as a function of the number of data points. It is assumed that the number of data points is not small, and that the true function cannot be exactly fit.

where ϵ is normally distributed with mean μ and standard deviation σ . Assume that we have an estimator $h(\mathbf{x}; \boldsymbol{\theta})$ trained by minimizing a cost function $\mathcal{C}(\mathbf{y}, h(\mathbf{x}))$ which we take to be the sum of squared errors:

$$\mathcal{C}(\mathbf{y}, h(\mathbf{x})) = \sum_i^n (y_i - h(\mathbf{x}_i; \boldsymbol{\theta}))^2.$$

Our best estimate for the model parameters:

$$\boldsymbol{\theta}_{\mathcal{D}} = \arg \min_{\boldsymbol{\theta}} \mathcal{C}(\mathbf{y}, h(\mathbf{x}; \boldsymbol{\theta})),$$

is a function of the dataset \mathcal{D} . If we imagine we have a set of datasets $\mathcal{D}_j = (\mathbf{y}_j, \mathbf{X}_j)$, each with n samples, we would like to calculate the expectation value of the cost function over all these datasets $E_{\mathcal{D}, \epsilon}$. We would also like to

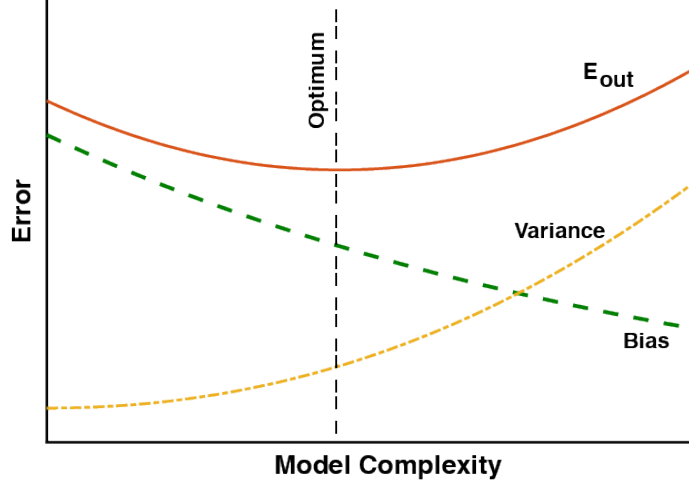


Figure 5.2: Typical out-of-sample error as a function of model complexity for a fixed dataset. Bias decreases monotonically with model complexity, while variance increases as a result of sampling noise.

calculate the expectation value over different instances of the noise ϵ . The expected generalization error can be decomposed as:

$$\begin{aligned} E_{\mathcal{D}, \epsilon}[\mathcal{C}(\mathbf{y}, h(\mathbf{X}; \boldsymbol{\theta}_{\mathcal{D}}))] &= E \left[\sum_i (y_i - h(\mathbf{x}_i; \boldsymbol{\theta}_{\mathcal{D}}))^2 \right] \\ &= \sum_i \sigma_{\epsilon}^2 + E_{\mathcal{D}}[(f(\mathbf{x}_i) - f(\mathbf{x}_i; \boldsymbol{\theta}_{\mathcal{D}}))^2]. \end{aligned} \quad (5.1)$$

The second term can be further decomposed as

$$\begin{aligned} E_{\mathcal{D}}[(f(\mathbf{x}_i) - f(\mathbf{x}_i; \boldsymbol{\theta}_{\mathcal{D}}))^2] \\ = (f(\mathbf{x}_i) - E_{\mathcal{D}}[h(\mathbf{x}_i; \boldsymbol{\theta}_{\mathcal{D}})])^2 + E[(h(\mathbf{x}_i; \boldsymbol{\theta}_{\mathcal{D}}) - E[h(\mathbf{x}_i; \boldsymbol{\theta}_{\mathcal{D}})])^2] \end{aligned} \quad (5.2)$$

The first term is what we have referred to as the bias:

$$\text{Bias}^2 = \sum_i (f(\mathbf{x}_i) - E_{\mathcal{D}}[h(\mathbf{x}_i; \boldsymbol{\theta}_{\mathcal{D}})])^2. \quad (5.3)$$

The bias measures the expectation value of the deviation of our model from the true function, i.e. the best we can do in the infinite data limit.

The second term is what we have referred to as the variance:

$$\text{Var} = \sum_i E[(h(\mathbf{x}_i; \boldsymbol{\theta}_{\mathcal{D}}) - E[h(\mathbf{x}_i; \boldsymbol{\theta}_{\mathcal{D}})])^2] \quad (5.4)$$

The variance measures the deviation of our model due to finite-sampling effects. Combining these effects we can decompose the out-of-sample error into:

$$E_{\text{out}} = \text{Bias}^2 + \text{Var} + \text{Noise}, \quad (5.5)$$

with $\text{Noise} = \sum_i \sigma_{\epsilon}^2$.

In general it can be much more difficult to obtain sufficient good data than to train a very complex model. Therefore it is often useful in practice to use a less complex model with higher bias, because it is less susceptible to finite-sampling effects.

5.0.3 Neural networks

Artificial Neural Networks (ANN) or Deep Neural Networks (DNN) are supervised learning models vaguely inspired by biological neural networks. The building blocks of neural networks are neurons that take a vector input of d features $\mathbf{x} = (x_1, \dots, x_d)$ and produce a scalar output $a(\mathbf{x})$. A neural network consists of layers of these neurons stacked together with the output of one layer serving as input for another. The first layer is typically known as the *input layer*, the middle layers as *hidden layers* and the final layer the *output layer*. The basic architecture is shown in figure 5.3. In almost all cases the output $a_i(\mathbf{x})$ of neuron i can be decomposed into a linear operation on the inputs passed through a non-linear activation function:

$$a_i(\mathbf{x}) = \sigma_i(z_i),$$

where σ_i is a non-linear function and z_i is the dot product between the inputs \mathbf{x} and a set of neuron-specific weights \mathbf{w}_i :

$$z_i = \mathbf{x}^T \mathbf{w}_i + b_i.$$

The term b_i is a neuron-specific re-centering of the input.

Typical choices of non-linearities/activation functions include the sigmoid and hyperbolic tangent functions, and Rectified Linear Units (ReLU). When the activation function is non-linear, the neural network with a single hidden

layer can be proven to be a *universal function approximator*, given an arbitrarily large number of neurons. We typically also want functions that are monotonic and smooth with a monotonic derivative.

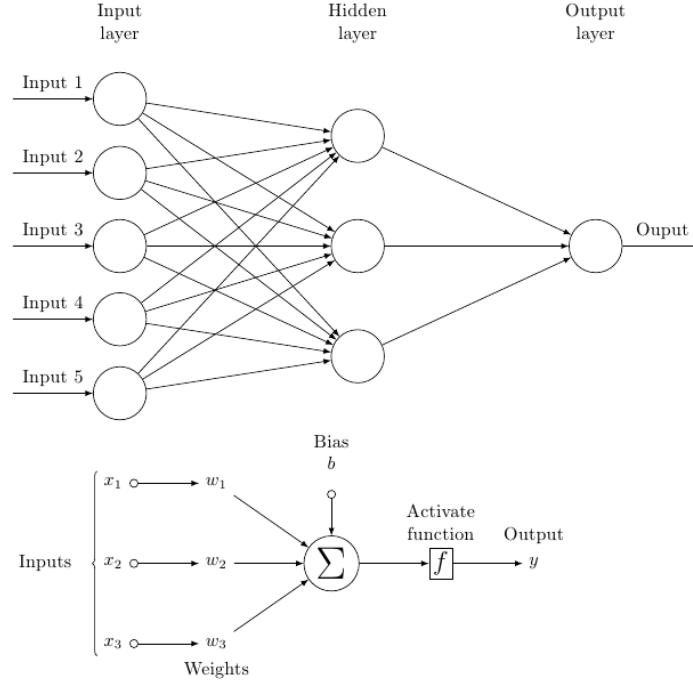


Figure 5.3: Text.

The simplest neural networks are known as *feed-forward* neural networks (FNN). The input layer is the vector \mathbf{x} of inputs, while each neuron in the first hidden layer performs a dot product between its weights \mathbf{w}_i and the inputs and passes it through a non-linearity σ_i . The activation function is typically shared across one or multiple layers $\sigma_i = \sigma$. The vector of neuron outputs \mathbf{a}_i serves as input to the next hidden layer until we reach the final layer. In the final layer the choice of activation function is dependent on the problem we are trying to solve. If we are performing non-linear regression the final activation function is often the identity $\sigma_i(z) = z$, or if we are doing classification the soft-max function is often employed.

Let \mathbf{x} be a vector of $d = 1, \dots, D$ inputs or *features*. Let $a_i^{(h)}$ denote the output of neuron $i = 1, \dots, N_d$ in layer $h = 1, \dots, H$. The output of neuron i in the first hidden layer $a_i^{(1)}$ is thus:

$$z_i^{(1)} = \mathbf{x}^T \mathbf{w}_i^{(1)} + b_i^{(1)}.$$

$$a_i^{(1)} = \sigma_i^{(1)}(z_i^{(1)}),$$

The inputs are iterated through each hidden layer until we reach the final layer:

$$\begin{aligned} z_i^{(H)} &= (\mathbf{a}_{H-1})^T \mathbf{w}_i^{(H)} + b_i^{(H)}. \\ a_i^{(H)} &= o_i = \sigma_i^{(H)}(z_i^{(H)}) \\ &= \sigma_i^{(H)} \left((\mathbf{a}_{H-1})^T \mathbf{w}_i^{(H)} + b_i^{(H)} \right) \\ &= \sigma_i^{(H)} \left(\left((\sigma_1^{(H-1)}, \dots, \sigma_{N_{H-1}}^{(H-1)}) \right)^T \mathbf{w}_i^{(H)} + b_i^{(H)} \right). \end{aligned} \tag{5.6}$$

This allows us to compose a complicated function $\mathbf{F} : \mathbb{R}^D \rightarrow \mathbb{R}^O$, with D the number of inputs and O the number of outputs. The *universal approximation theorem* tells us that this simple architecture can approximate any of a large set of continuous functions given appropriate choice of weights \mathbf{w}_i^h and mild assumptions on the activation functions. The theorem requires only a single hidden layer, where the strength of the approximation relies on the number of neurons. In practice it has been found that adding more layers produces faster convergence and higher accuracy, which has given rise to the field of *deep learning*.

5.0.4 Backpropagation

Given a set of datapoints (\mathbf{x}_i, y_i) , $i = 1, \dots, n$, the value of the cost function is entirely determined by the weights and biases of each neuron in the network. We define learning narrowly as adjusting the parameters of the network in order to minimize the cost function.

Gradient descent is a simple, but powerful method of finding the minima of differentiable functions. Given a function $F : \mathbb{R}^d \rightarrow \mathbb{R}$, and an initial value \mathbf{x}_0 we define an iterative procedure:

$$\mathbf{x}_{n+1} = \mathbf{x}_n - \eta \nabla F(\mathbf{x}_n),$$

where η is known as the *learning rate*. The procedure terminates when the norm $|\nabla F(\mathbf{x}_n)|$ or $|x_{n+1} - x_n|$ is appropriately small.

The learning rate is not necessarily fixed throughout the procedure, and proves crucial to the convergence of the method. If f is convex, and η is reasonably small, convergence is guaranteed. Convergence may be very slow however, and if f is not convex you are only guaranteed to find local minima, and this makes the method very sensitive to initial conditions.

In order to train the model, we need to calculate the derivative of the cost function with respect to a very large number of parameters multiple times. However, numerical calculation of gradients is very time consuming. The *backpropagation* algorithm is a clever use of the chain rule that allows us to calculate gradients efficiently.

Assume that there are L layers in our network with $l = 1, 2, \dots, L$ indexing the layers, including the output layer and all the hidden layers. Let w_{ij}^l denote the weight for the connection from the i -th neuron in layer $l - 1$ to the j -th neuron in layer l . Let b_j^l denote the bias of this j -th neuron.

The activation a_j^l of the j -th neuron in the l -th layer is related to the activities of the neurons in the layer $l - 1$ by:

$$a_j^l = f \left(\sum_i w_{ij}^l a_i^{l-1} + b_j^l \right) = f(z_j^l),$$

where f is some activation function.

The cost function \mathcal{C} depends directly on the activations in the output layer, and indirectly on the activations in all the lower layers. Define the error Δ_j^L of the j -th neuron in the L -th (final) layer as the change in cost function with respect to the weighted input z_j^L :

$$\Delta_j^L = \frac{\partial \mathcal{C}}{\partial z_j^L}.$$

Define analogously the error Δ_j^l of neuron j in the l -th layer as the change in cost function with respect to the weighted input z_j^l :

$$\Delta_j^l = \frac{\partial \mathcal{C}}{\partial z_j^l}.$$

This can also be interpreted as the change in cost function with respect to the bias b_j^l :

$$\Delta_j^l = \frac{\partial \mathcal{C}}{\partial z_j^l} = \frac{\partial \mathcal{C}}{\partial b_j^l} \frac{\partial b_j^l}{\partial z_j^l} = \frac{\partial \mathcal{C}}{\partial b_j^l},$$

since $\partial b_i^j / \partial z_j^l = 1$.

The error depends on neurons in layer l only through the activation of neurons in layer $l + 1$, so using the chain rule we can write:

$$\begin{aligned} \Delta_j^l &= \frac{\partial \mathcal{C}}{\partial z_j^l} = \sum_i \frac{\partial \mathcal{C}}{\partial z_i^{l+1}} \frac{\partial z_i^{l+1}}{\partial z_j^l} \\ &= \sum_i \Delta_i^{l+1} \frac{\partial z_i^{l+1}}{\partial z_j^l} \\ &= \sum_i \Delta_i^{l+1} w_{ij}^{l+1} f'(z_j^l) \\ &= \left(\sum_i \Delta_i^{l+1} w_{ij}^{l+1} \right) f'(z_j^l). \end{aligned} \tag{5.7}$$

The sum comes from the fact that any error in neuron j in the l -th layer propagates to all the neurons in the layer $l + 1$, so we have to sum up these errors.

This gives us the equations we need to update the weights and biases of our network:

$$\begin{aligned} \frac{\partial \mathcal{C}}{\partial w_{ij}^l} &= \frac{\partial \mathcal{C}}{\partial z_j^l} \frac{\partial z_j^l}{\partial w_{ij}^l} = \Delta_j^l a_i^{l-1}. \\ \frac{\partial \mathcal{C}}{\partial b_j^l} &= \Delta_j^l. \end{aligned}$$

Now, if we have the error of every neuron j at the output layer, Δ_j^L , equation ?? gives us the recipe for calculating the error in the preceding layer until we reach the first hidden layer, and we are done. All we are missing is the error at the output layer:

$$\Delta_j^L = \frac{\partial \mathcal{C}}{\partial z_j^L} = \frac{\partial}{\partial z_j^L} \left(- \sum_c y_c \log f(z_c^L) \right),$$

where f is the softmax function. Taking the derivative of each term:

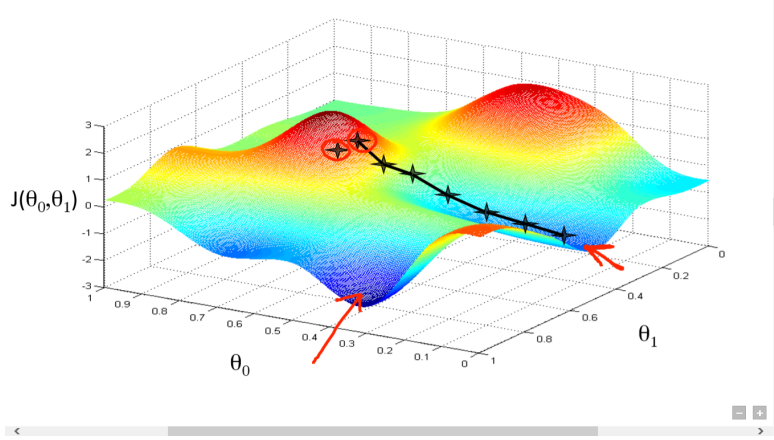


Figure 5.4: Text.

$$\frac{\partial \log f(z_c^L)}{\partial z_j^L} = \frac{\partial}{\partial z_j^L} \left(z_c^L - \log \left(\sum_c \exp(z_c^L) \right) \right) = \delta_{jc} - f(z_j^L),$$

where δ_{jc} is the Kronecker-Delta. This gives us the final expression we need:

$$\begin{aligned} \Delta_j^L &= - \sum_c y_c (\delta_{jc} - f(z_j^L)) \\ &= - \sum_c y_c \delta_{jc} + \sum_c y_c f(z_j^L) \\ &= -y_j + f(z_j^L) \sum_c y_c \\ &= f(z_j^L) - y_j \\ &= \hat{y}_j - y_j. \end{aligned} \tag{5.8}$$

5.0.5 Optimization

Gradient descent, momentum, ADAM, initialization, batching, vanishing, normalization (?).

Part II

Implementation

Chapter 6

Deep potential molecular dynamics

Chapter 7

Quantum Espresso and I-PI

Chapter 8

Pytorch

Chapter 9

LAMMPS

Part III

Results

Chapter 10

Empirical potentials

Chapter 11

Hydrogen systems

Chapter 12

Conclusion

Hello, here is some text without a meaning. This text should show what a printed text will look like at this place. If you read this text, you will get no information. Really? Is there no information? Is there a difference between this text and some nonsense like “Huardest gefburn”? Kjift – not at all! A blind text like this gives you information about the selected font, how the letters are written and an impression of the look. This text should contain all letters of the alphabet and it should be written in of the original language. There is no need for special content, but the length of words should match the language.

This is the second paragraph. Hello, here is some text without a meaning. This text should show what a printed text will look like at this place. If you read this text, you will get no information. Really? Is there no information? Is there a difference between this text and some nonsense like “Huardest gefburn”? Kjift – not at all! A blind text like this gives you information about the selected font, how the letters are written and an impression of the look. This text should contain all letters of the alphabet and it should be written in of the original language. There is no need for special content, but the length of words should match the language.

And after the second paragraph follows the third paragraph. Hello, here is some text without a meaning. This text should show what a printed text will look like at this place. If you read this text, you will get no information. Really? Is there no information? Is there a difference between this text and some nonsense like “Huardest gefburn”? Kjift – not at all! A blind text like this gives you information about the selected font, how the letters are written and an impression of the look. This text should contain all letters of

the alphabet and it should be written in of the original language. There is no need for special content, but the length of words should match the language.

After this fourth paragraph, we start a new paragraph sequence. Hello, here is some text without a meaning. This text should show what a printed text will look like at this place. If you read this text, you will get no information. Really? Is there no information? Is there a difference between this text and some nonsense like “Huardest gefburn”? Kjift – not at all! A blind text like this gives you information about the selected font, how the letters are written and an impression of the look. This text should contain all letters of the alphabet and it should be written in of the original language. There is no need for special content, but the length of words should match the language.

Hello, here is some text without a meaning. This text should show what a printed text will look like at this place. If you read this text, you will get no information. Really? Is there no information? Is there a difference between this text and some nonsense like “Huardest gefburn”? Kjift – not at all! A blind text like this gives you information about the selected font, how the letters are written and an impression of the look. This text should contain all letters of the alphabet and it should be written in of the original language. There is no need for special content, but the length of words should match the language.

Appendices

Appendix A

Title 1

Appendix B

Title 2

Bibliography

- [1] Albert Einstein. “Zur Elektrodynamik bewegter Körper. (German) [On the electrodynamics of moving bodies]”. In: *Annalen der Physik* 322.10 (1905), pp. 891–921. DOI: <http://dx.doi.org/10.1002/andp.19053221004>.