MPhys Project - Machine Learned Potentials For MD Simulations

University of Exeter EMPS

Natan Szczepaniak

Supervisor: Prof. Saverio Russo

Week 1 (26/05/20-29/05/20)

Contents

Day 1 - 26/05/20

1.1 Project Restructure

As our previous project aim "Neuromorphic Computing for Artifical Intelligence" was scrapped due to the breakout of the pandemic preventing us accessing the laboratories, we had to find a new topic. When this change took place our supervisor changed the aim of our project to.

> "Develop a neural network code applied to simple physics problems"

After a meeting with the other project members we concluded that this is extremely broad and it is hard to find a specific field to focus on. We decided to take initiative to find which areas of physics could be improved by methods of machine learning algorithms. We gathered a list of the following:

- Nuclear and High Energy Physics
- Particle Physics
- Quantum Physics
- Astrophysics
- Condensed Matter Physics

Machine Learning algorithms are only useful when paired with a large database. Having considered our options and doing brief preliminary research into these fields we came up with three areas of focus. Nuclear Physics research at the CERN uses machine learning (https://cerncourier.com/a/the-rise-of-deep-learning/) to reconstruct and classify particle collisions. Astrophysics research uses ML for a variety of tedious tasks ranging from noise reduction to classification of astronomical events. Lastly, in Condensed Matter Physics there is a subfield of using machine learning algorithms to fit new potential energy surfaces which can be used in molecular dynamics simulations.

After a discussion with the group members we agreed that we would all be the most interested in exploring the niche field of generating accurate potential energy surfaces for dynamics simulations as it gives us a lot to work with. Another reason for our choice is that our project leader specialises in condensed matter physics meaning we can get support from him as well as contacts to other researchers in the field. Additionally, the problem at hand would be solved by a supervised learning algorithm which means we can compare our model to preexisting data and look for discrepancies.

Once the area of interest was confirmed between the group members, we set up a meeting with our project leader and started independent research on the subject. I compiled a list of papers which cover this sub-field of physics to discuss.

[1] **Machine learning for interatomic potential models** https://aip.scitation.org/doi/10.1063/1.5126336 (https://aip.scitation.org/doi/10.1063/1.5126336)

[2] **Generalized Neural-Network Representation of High-Dimensional Potential-Energy Surfaces** http://cacs.usc.edu/education/cs653/Behler-NNPES-PRL07.pdf (http://cacs.usc.edu/education/cs653/Behler-NNPES-PRL07.pdf)

[3] **Machine Learning Potentials for atomistic simulations** https://aip.scitation.org/doi/full/10.1063/1.4966192 (https://aip.scitation.org/doi/full/10.1063/1.4966192)

[4] **Machine Learning for Atomic Forces in a Crystalline Solid: Transferability to Various Temperatures**https://arxiv.org/abs/1608.07374 (https://arxiv.org/abs/1608.07374)

[5] **Machine learning for quantum mechanics in a nutshell** https://onlinelibrary.wiley.com/doi/full/10.1002/qua.24954 (https://onlinelibrary.wiley.com/doi/full/10.1002/qua.24954)

[6] **An Investigation of Machine Learning Methods Applied to Structure Prediction in Condensed Matter** https://arxiv.org/pdf/1405.3564v1.pdf (https://arxiv.org/pdf/1405.3564v1.pdf)

[7] **Gassian approximation potentials: The accuracy of quantum mechanics, without the electrons** (Haven't yet read but looks very useful for descriptor type networks) https://arxiv.org/pdf/0910.1019.pdf (https://arxiv.org/pdf/0910.1019.pdf)

[8] **Newton vs the machine: solving the chaotic three-body problem using deep neural networks:** https://arxiv.org/pdf/1910.07291.pdf (https://arxiv.org/pdf/1910.07291.pdf)

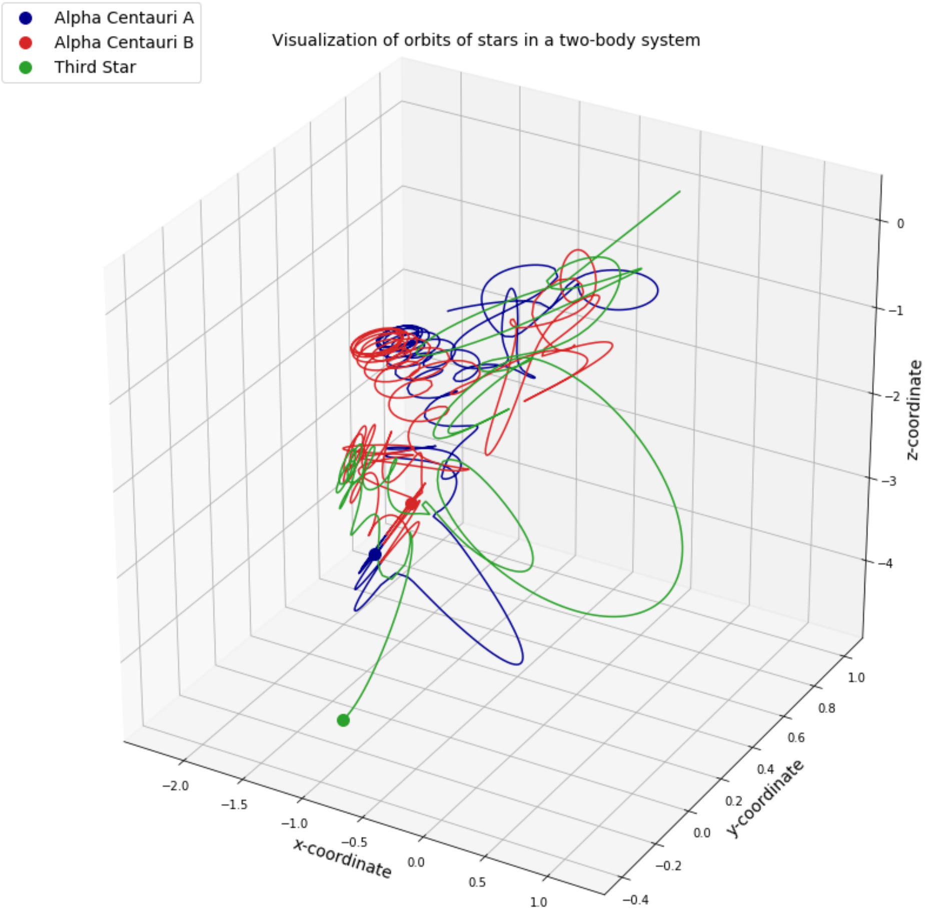
[9] **Constructing exact representations of quantum many-body systems with deep neural networks (Advanced)** https://medium.com/nieuwsgierigheid/machine-learning-quantum-physics-27e316d4ed77 (https://medium.com/nieuwsgierigheid/machine-learning-quantum-physics-27e316d4ed77)

1.2 Molecular Dynamics Simulations

The potential energy surface model derived from the statistical learning model will then be plugged into a Molecular Dynamics simulator. This most likely be written in a more low level language such as C for efficiency.

We plan on using a premade MD simulator as making one from scratch would not be viable and would not match the quality and accuracy of contemporary software. This would mean we would have to work to integrate our machine learned algorithm together with software that would be able to use it. This would mean possibly having to output our potential energy surface as a file compatible with such a program. (LAMMPS potential candidate)

To get familiarised with Python however, (The language we will be using for the majority of this project) and computer many body simulations we looked at an extremely simple 3 body simulation. (https://towardsdatascience.com/modelling-the-three-body-problem-in-classical-mechanics-using-python-9dc270ad7767) that simulates the time evolution of a 3 star system which inputs the properties of each star (mass, coordinates) and outputs the evolution from which the properties of the entire system can be obtained. For this to run I had to change the scipy sci.array() arrays into the more commonly used numpy np.array() arrays. This only shows how quickly the field of programming is moving and how versioning is going to be very important to pay attention to in the future.



1.3 Software

Our language of choice for this project will be Python 3.7 due to the familirarity, high accessability and large number of libraries dedicated for data analysis. For collaborative work we chose to use voice chat (Discord) for daily communication combined with Google Drive for sharing files. For electronic lab books we decided to use Jupyter notebook as we are able to combine Markdown text, HTML components, LateX Markup, Python Code as well as images in one file that can be exported as a .pdf file for submission.

Python is also a fairly easy language to learn as it includes various libraries for data analysis such as Numpy, Pandas, SciKitLearn, TensorFlow, matplotlib, etc. which are all being used in industry and research today.

1.4 [1] Machine learning for interatomic potential models

> *"Machine-learned interatomic potentials have demonstrated excellent accuracy compared to the methods used to train them and hold the promise of accelerating aspects of atomic-scale computational research by orders of magnitude."*

This paper was a good introduction to the field of empirically dervied potential energy surfaces for MD simulations. It highlights that exact solutions to the Schrödinger equation for real world systems do not exist. Instead density functional theory (DFT) is used to computationally approximate the solution.

> *"Such physics-derived interatomic potentials, which are sometimes referred to as "classical" or "empirical" interatomic potential models, are often used to model systems at particularly large time or length scales. "*
>
> *"Ercolessi and Adams demonstrated that this can be effectively done by fitting the potential to DFT-calculated forces, as a single density functional theory calculation provides both the energy of a given configuration and, with little extra computational cost, the forces on each of the atoms, generating a total of 3N + 1 points of training data (before accounting for symmetry constraints) for a system with N atoms.*

The data generated by the approximate method, will then be passed down to the machine learning algorithm which will be used to fit the potential energy surface.

> *"In supervised machine learning, the objective is to identify a function f that accurately predicts values y from sets of input data x. In the context of interatomic potential models, x represents the atomic species and nuclear coordinates, y is the value on the potential energy surface, and f is the interatomic potential model to be learned."*

This will require a lot of data. This paper (https://aip.scitation.org/doi/10.1063/1.5126336) suggest using data from DFT (http://newton.ex.ac.uk/research/qsystems/people/coomer/dft_intro.html) calculations to form and adjust a potential energy surface. Possible data sets include the OQMD (https://gist.github.com/jallen30gt/b9e747e7424d643dd441) database which has a dedicated python library for accessing, Material Project Database (https://www.materialsproject.org/docs/api) which has a specialised API for data access and the ANI-1 Database (https://github.com/isayev/ANI1_dataset). This is extremely important as with majority of machine learning tasks, handling the data is a cumbersome task and using an API helps us only request and use data that concerns us making the process more efficient.

When choosing a supervised learning algorithm for our purpose we must consider three fundamental steps outlined in the paper:

1. Choosing a hypothesis space by choosing constraints. (eg. Choose a finite set of basis functions to optimize instead of the entire potential energy surface.)
2. Determine the objective function (such as squared error)
3. Determine method (eg. neural network, logistic regression, SVM etc.)

Another three things to consider when developing potential models for materials are:

1. Biggest contribution to the potential energy surface is from atoms that are in the cutoff radius of around 4-6$\mathring{A}$ meaning we can introduce a maximum distance at which the contribution is cut off to save on computational cost.
2. Potential energy must be invariant under permutations
3. The potential energy surface must be continuous and smooth

The most popular approach to these conventions in the field is implementing "constructing descriptors" or "fingerprints". This is what the potential energy surface looks like in the local environment around the atom inside the cutoff radius. Which are invariant to permutations from surrounding atoms. By defining these "fingerprints" we give the machine learning algorithm a desired output which is optimised using the objective function. Types of descriptors used in the field include:

- Atom-centered symmetry functions
- Bispectrum components
- Coulomb Matrices
- Smooth overlap of atomic positions

This paper outlines previous methods and approaches that utilise these descriptors including the Behler-Parrinello approach, Gaussian Approximation potentials (GAP) and spectral neighbor analysis potentials (SNAP). For the beginning of our project we could potentially attempt to recreate one of these methods and look at the benefits and drawbacks of using these methods. As this is a generally new field, new methods are currently being researched. The field itself however, relies heavily on the work of Behler and Parrinello in 2007:

https://journals.aps.org/prl/abstract/10.1103/PhysRevLett.98.146401 (https://journals.aps.org/prl/abstract/10.1103/PhysRevLett.98.146401)

when the field was starting to pick up its pace due to the increase in available data caused by advances in computing speed. The method proposed in this paper was interestingly implemented by a Harvard post graduate student into a fully functional tool "ænet".

http://ann.atomistic.net/ (http://ann.atomistic.net/)

This tool compromises of a C and Fortran libraries that "can be integrated in existing simulation software to actually use ANN potentials in atomistic simulations". This utilises the original approach proposed by Behler however a plethora of different approaches as the field of machine learning grows by the day. Personally, I think it would be interesting to implement my favourite algorithm Long-short-term-memory LSTM adjusted to work in a supervised fashion using gradient descent as I haven't seen this attempted before.

Day 2 - 27/05/20

Back to the paper.

The paper talks about 3 emerging methods that are being used currently. Moment Tensor Potentials, Message-passing networks and symbolic regression. I also found a github repository which includes implementation of all of these 3. https://github.com/materialsvirtuallab/mlearn/blob/master/docs/install.md (https://github.com/materialsvirtuallab/mlearn/blob/master/docs/install.md)

1.4.1 Moment Tensor Potentials

Developed by Shapeev (http://epubs.siam.org/doi/10.1137/15M1054183) the moment tensor approach includes expressing the potential energy as a "linear combination of polynomial basis functions representing one-body, two-body and three-body interactions so they are analogous to cluster expansions". One difference is that the basis is defined over continuous atomic positions and not discrete sites.

Potential energy is a sum of each atom's contribution:

$$V(\mathbf{R},\mathbf{z}) = \sum_{i=1}^{N} V_{local}(\mathbf{R}-\mathbf{r}_i,\mathbf{z},z_i)$$

where V_{local} is the contribution from each atom at $\mathbf{r}_i$ of species z_i. $\mathbf{R} = (\mathbf{r}_1, \mathbf{r}_2, \mathbf{r}_3, \dots)$ and $\mathbf{z} = (z_1, z_2, z_3, \dots)$. For this method, V_{local} is expanded as a linear combination of basis functions B_α

$$V_{local}(\mathbf{R}-\mathbf{r}_i,\mathbf{z},z_i) = \sum_{\alpha} V_\alpha B_\alpha(\mathbf{R}-\mathbf{r}_i,\mathbf{z},z_i)$$

α is a symmetric square matrix of non-negative integers and V_α is a linear coefficient. The basis function can be constructed using a square matrix of a given size (paper uses 3 however this can be adjusted). The paper describes a matrix such as

$$\beta = \begin{pmatrix} f_{\alpha_{11}}(r_{ij},z_i,z_j) & (\mathbf{r}_{ik}\cdot\mathbf{r}_{ij})^{\alpha_{12}} & (\mathbf{r}_{il}\cdot\mathbf{r}_{ij})^{\alpha_{13}} \\ (\mathbf{r}_{ij}\cdot\mathbf{r}_{ik})^{\alpha_{21}} & f_{\alpha_{22}}(r_{ik},z_i,z_k) & (\mathbf{r}_{il}\cdot\mathbf{r}_{ik})^{\alpha_{23}} \\ (\mathbf{r}_{ij}\cdot\mathbf{r}_{il})^{\alpha_{31}} & (\mathbf{r}_{ik}\cdot\mathbf{r}_{il})^{\alpha_{32}} & f_{\alpha_{33}}(r_{il},z_i,z_l) \end{pmatrix}$$

In order to make sure our potential goes to 0 at a cutoff radius, we use f_μ which limits the contribution at a cutoff distance. The basis function is then given by:

$$B_{alpha}(\mathbf{R}-\mathbf{r}_i,\mathbf{z},z_i) = \sum_{j,k,l} (\Pi_{a,b\leq a}\beta_{ab})$$

It is important to understand that this paper defines the vectors $\mathbf{r}_{if} = \mathbf{r}_f - \mathbf{r}_i$. A 3x3 matrix such as this one, we can find a basis function for a 4 body interaction but this scales such as if you had a NxN matrix you could describe a basis of (N+1) body interactions. This includes atoms i,j,k and l. This system is set up so that basis functions are invariant of rotations and permutations from neighbouring atoms as the function only depends on the radial distances. Shapeev also realised computational cost can be saved by rewriting the above basis function as:

$$M_{\mu,\nu} = \sum_j f_\mu(r_{ij},z_i,z_j)\mathbf{r}_{ij}^{\otimes\nu}$$

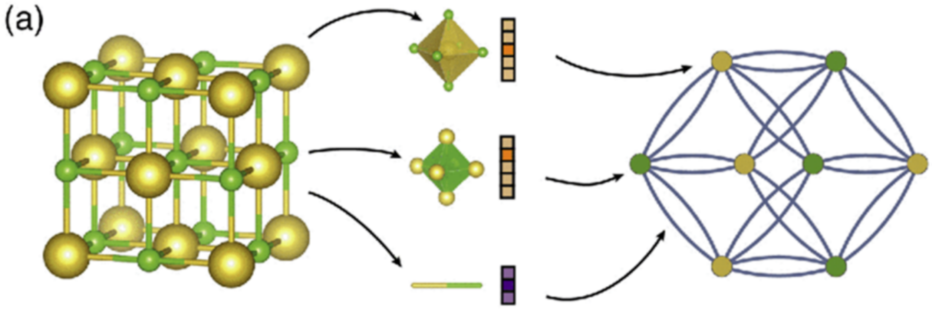
where $\mathbf{r}^{\otimes\nu}$ means taking the tensor product of $\mathbf{r}_{ij}$ ν times. The paper outlines that this method has been proven to have a better speed/accuracy tradeoff than GAP, neural network and SNAP potentials making it an area of interest for our project for possible reconstruction. This method fits potentials to a set of energy forces and stresses calculated via DFT. Additionally, for systems of a singular species of atom, "the potential is linear with respect to the unknown parameters" meaning simple linear regression can be applied. For systems of several different species a different optimisation algorithm must be used. As of the publication of the paper (January 2020), it mentions that a Bayesian approach in determining the regularisation parameters has not been attempted giving us our first possible gateway into the project in with the possibility of creating something unique and new. Firstly however, we will try to find source code for a prexisting implementation in which we can learn on.

An implementation of this code is available in the github link above however the gitlab page the code is stored on is access on request only so we will perhaps need to contact Alexander Shapeev about this. http://www.shapeev.com/ (http://www.shapeev.com/) From his history he has a lot of other publications worth looking at for future reference.

1.4.2 Message Passing Networks

This approach uses graph networks. This means the topology of the NN is based on the topography of the atomic structure itself. Atoms are represented as nodes on a network and atoms are connected by edges which are represented as connections between the nodes. This method has been used for molecules but in theory could work for crystalline structures. This looks like it would require precise modelling of a network which could possibly be done using TensorFlow which allows to do just that easily. This however would mean that it would be more difficult to create a general code as a new network would have to be created for each new structure.

The paper gives a rather nice visualisation of this.



As seen from the image the topography of the structure determines the structure of the network with nodes and edges. This method is called "Message Passing" in which information is passed from one node to the other however this is analogous to a Convolutional Neural Network with multiple layers. Each node has an assigned a feature vector $\mathbf{n}_i^{(t)}$ and every edge has a feature vector $\mathbf{e}_{ij}^{(t)}$. i and j represent the atom and t is the number of iterations. $\mathbf{n}_i^{(0)}$ would contain information about the atom and $\mathbf{e}_{ij}^{(0)}$ would contain information about the distance.

The nodes exchange messages indicated by $\mathbf{m}_i^{(t)}$ which is a function of $\mathbf{n}_i^{(t)}$, $\mathbf{e}_{ij}^{(t)}$ and $\mathbf{n}_j^{(t)}$

$$\mathbf{m}_i^{(t)} = \sum_j m(\mathbf{n}_i^{(t)}, \mathbf{e}_{ij}^{(t)}, \mathbf{n}_j^{(t)})$$

where m is the learned information to be found. This message the updates the node after a time iteration:

$$\mathbf{n}_i^{(t+1)} = u(\mathbf{n}_i^{(t)}, \mathbf{m}_i^{(t)})$$

where u is learned information (In standard ML NNs this would be called the "weight"). Each one of the nodes sends out a message and after each time iteration it affects more and more nodes around it like a "ripple effect". Only thing is that all of these nodes do the same simultaneously. After some iterations, the desired property can be learned from the system

This approach is quite significantly different to the previous approach using descriptors as it does not have a cut-off radius and does not compute many body calculations. Instead, it takes into account 2 interaction at a time. This is both a blessing and a curse as you cannot use the same network for different systems but it is quite compuationally efficient when designed for one specfic one and scales well. Using this method we can predict different material properties including our desired potential energy surface. There exist different types of Message passing networks for different systems simple and complex. The parameters of n amd e are left to interpretation and creativity. In order to get a potential energy surface out however, we need to include distance in the e feature vector.

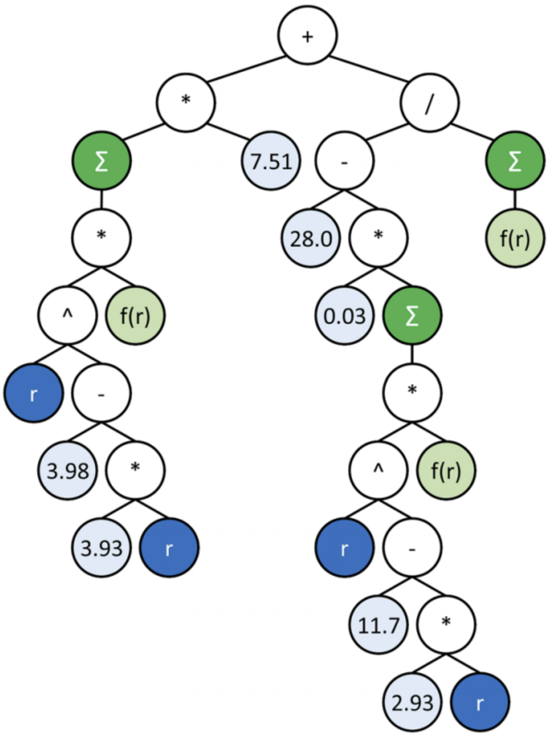
An example of a message-passing network called a deep tensor neural network (DTNN), this network has been observed to Create continuous Potential Energy surfaces (https://doi.org/10.1038/ncomms13890). This is able to create potential energy surfaces for extremely complicated organic molecules with reasonably high accuracy. From my understanding, this algorithm takes data from MD simulations instead of DFT.

This is quite a quirky method and unless we decide to delve deep into neural network architectures I highly doubt we will be pursuing a system like this however it is good to know it is a possiblity.

1.4.3 Symbolic Regression

The last approach highlighted by the paper is Symbolic Regression. In this method, we create a hypothesis space of different simple mathematical expressions structures like a tree which is searched using an algorithm to find the right one. To search the hypothesis space, a genetic algorithm would be used to find the parameters for interatomic potential models. This approach has been used to rediscover the well known Lennard-Jones Potential as well as a simplified 3-body Stillinger-Weber potential. Kenoufi and Kholomurodov created a potential for an argon dimer using DFT data (A. Kenoufi and K. Kholmurodov, “Symbolic regression of inter-atomic potentials via genetic programming,” in 7th Russian-Japanese International Workshop MSSMBS’14, Molecular, 2014.).

Interestingly I personally came across genetic algorithms in the previous project we were looking at before the restructure. Genetic algorithms tackle optimisation problems. In general terms, the start off with a random population (sets of parameters). After every iteration of the algorithm we assign each person a score called "fitness". We sort the population by their "fitness", delete the bottom half and let the top half "reproduce" by mixing their details (parameters). The process is repeated until the "fitness" threshold (desired fitness score). This can be implemented to find the perfect function of a potential energy surface for a given system. Unlike the previous method it is very generalised and can be applied to many systems using only one model. The image below shows the hypothesis space tree to be explored by an algorithm like this



The authors of this paper created an open source package called Potential Optimization by Evolutionary Techiques (POET). This would be very helpful in the aim of understanding someone elses source code before attempting our own task. This is quite a simple algorithm yet rather powerful. The simplicity of the method allows for high speed and little data to be used.

This would be a very attractive method for us to try due to its simplicity, potential (no pun intended) and accessibility. Additionally, the authors mention possible improvements that can be made to this method in order to reduce the uncertainty in potential model development.

> *There are also a number of opportunities to explore algorithmic improvements for more efficiently and reliably identifying good interatomic potentials. In particular, it would be beneficial to explore approaches to symbolic regression that are less stochastic than current approaches, reducing the uncertainty in potential model development.*

2.1 Discussion

Out of these three methods the probability of us attempting each one of them would be ranked (most likely to least likely) as:

1. Symbolic Regression (Descriptor-type)
2. Moment Tensor Potentials
3. Message Passing Networks

Generally, this paper was an extremely good start to the field as it outlines 3 different methods of how machine learning algorithms are applied to finding potential energy surfaces for different structures.

We explored Simple Regression algorithms, Convolutional Neural Networks as well as genetic algorithms. Due to the large variety of parameters that can be explored and the number of supervised learning algorithms available we can choose a combination that we can use for our unique project. There is also a lot of space for extending our project further if time allows. We expect that familiarising ourselves with the code will help us understand the space of ML in Condensed matter research more however ultimately we will focus on a Descriptor based method such as Moment Tensor Potentials in the long run.

The only things this paper did not clarify very well is the cutoff function f_μ for the "fingerprints" in the MTP method and the actual application of these methods. Next, we will study the paper "Machine Learning Potentials for atomistic simulations" as after skimming it, it seems to have more detail.

2.2 Meeting with Supervisor

After sending the following email to our supervisor, we arranged a Microsoft Teams meeting with our supervisor to discuss our finding and run our ideas by him.

> Dear Saverio,
>
> Hope you are doing well during these difficult times. Given that it is now officially the beginning of the week of our MPhys > Project research, I had a couple questions I wanted to ask regarding aims of the project.
>
> Shortly after exams, Max, Nathan and I got together to discuss how we should go about the project now that our previous work has been rendered irrelevant. We all agreed that "Develop a neural network code applied to simple physics problems" is extremely broad and it is hard to find a specific field to focus on.
>
> Considering the focus of our former project being condensed matter, we have looked into how machine learning can be applied to this field and have found several papers on the topic. What we propose is using machine learning models in order to create empirical interatomic potentials for molecular dynamics simulations. (Papers considered have been attached).
>
> We tried choosing a topic that:
>
> - Isn’t too trivial giving substance to the project
> - Isn’t too complex so we are not overwhelmed
> - Is related to your field of research
>
> We have also decided and set up our preferred method of an electronic notebook in the form of a Jupyter notebook which allows us to include Code, Markup text and LateX equations which can be exported as a pdf file.
>
> What we want to ask is what exactly is expected from us at the end of these three weeks considering our research phase was scrapped entirely. We have slowly started working on the direction of our project however we will require more guidance on this.
>
> Another thing we want to ask is who should we go to for guidance regarding programming and general knowledge on the subject if we get stuck.
>
> If possible, we would all be interested in having a call this week considering the tight deadlines and intensive workload as we are not certain whether we can go through with this idea without a green light from a supervisor. Preferably tomorrow would be ideal as we have a few ideas we want to run by you and figure out if they are viable.
>
> Thanks,
>
> Natan Szczepaniak

In the meeting, we discussed the project idea. Our supervisor firstly suggested that because Medicinal Science is a big field of interest with a lot of funding we should perhaps take our project into a different direction. We agreed it would be a good idea however, we wouldnt feel confident stepping into an unknown field with our time restrictions and decided to stick with our original idea.

I personally thought that this does tie into medicine at a bit of a strech as MD simulations are used by chemists developing chemicals and pharmaceuticals as well as involving more physics. Saverio agreed and said we have his full support on this. He mentioned that after the 3 weeks we would be expected to have looked at an implementation of these methods in a code and try to fully understand and annotate it however also instructed us to contact Matthew Bates on this.

Here are the notes taken during the meeting:

- Project idea change
- Biomedicine (We have no experinece and knowledge about medicine as neither of us have taken any biophysics modules and don't take particular interest in it either)
- Sticking with Machine Learned potentials and we have our supervisors support
- DFT discussion
- By the end of 3 week we are expected to reasearch and understand an implementation of this where its been done before
- Email Saverio with any questions

- Next meeting next monday 01/06/2020

2.3 Density Functional Theory (DFT)

Another thing discussed in the meeting was Density Functional Theory (DFT). I brought it up as it is an area of physics our supervisor would be familiar with and we would need to understand for our project. Prof. Russo gave us a resource to a book about DFT (https://www.amazon.co.uk/Solid-State-Physics-Giuseppe-Grosso/dp/0123850304/ref=sr_1_1?dchild=1&keywords=solid+state+physics+grosso&qid=1590575222&sr=8-1). In chapter 4.8 of the book, it explains DFT to be an approximation method for many body systems. This approximation is necessary for calculating ground state properties of complex systems due to the computational demand of using the well known method of solving the Schrodinger equation.

Standard method falls short for complex structures as the number of degrees of freedom increases. A popular example that used to describe this from what i have read is a silicon electronic structure. There are 28 electrons in the unit cell of silicon, in 3D this makes the number of coordinates needed 84 (28x3), in addition to the spin states of the electrons (1 per electron) this gives us 112 coordinates for a simple silicon unit cell.

According to the calculations done in this video (https://www.youtube.com/watch?v=zH_qF6oG82U) explaing the topic further, it would take the age of the universe to calculate an integral over the degrees of freedom if we discretise each dimension with 10 points on a high performace computer. Clearly this is not viable given our current technology.

Density Functional Theory approaches solving many body systems using only the electron density reducing the amount of calculations that need to be carried out. The theory claims that all the information about the ground state of the system is contained within the simple 3 dimensional electron density function.

This is a big topic in the field of condensed matter physics and requires more research, for now we know that Born-Oppenheimer and mean field approximations are the key concepts that lead to the Hartree & Hartree Fock equations which are essential to know before attempting full DFT.

We will judge our level of understanding as our project develops.

For future exercise: http://dcwww.camd.dtu.dk/~askhl/files/python-dft-exercises.pdf (http://dcwww.camd.dtu.dk/~askhl/files/python-dft-exercises.pdf) </span>

Useful Python Library Atomic Simulation Environment (ASE) python library

2.4 Roadmap

I contacted a PHD collegue of mine that has conducted similar research in the field in the past. I asked him a few questions to get insight into this niche subfield of physics.

Together we came up with a general workflow roadmap to follow that is common among different approaches as to not get lost.

1. Obtain DFT calculation data or Run personal DFT simulations (computationally intensive)
2. Choose a data descriptor, make a tool to compute the descriptor from DFT data
3. Train a model
4. Fit a potential
5. Run MD with the potential

He also suggested that for Molecular Dynamics Simulations for this purpose, LAAMPS would the the software of choice in contemporary research. This allows us to focus on the machine learning part of the project as the communication between LAAMPS and python can be established.

We agreed that starting with a single species structure for the first part of the project would be ideal as it would help create simpler code on which we can improve on to work for multi speciees structures.

He suggested a paper by Suzuki Miyazaki on the topic. "Machine Learning for Atomic Forces in a Crystalline Solid: Transferability to Various Temperatures". https://arxiv.org/abs/1608.07374 (https://arxiv.org/abs/1608.07374).

After speaking to the other group members we agreed that we will most likely work with silicon due to its simple structure and well established data. Then once this is done, we can move onto more complicated systems. For now, we will follow the workflow roadmap and apply it to any new things as we learn them. Because our previous project studied the structure of HfO2 and MoS2, we thought we may try find a potential surface for these compounds as a challenge.

Day 3 - 28/05/20

3.1 [2] Generalized Neural-Network Representation of High-Dimensional Potential-Energy Surfaces

3.1.1 Approach

Jörg Behler is a very prominent figure in this field. After coming across his 2016 paper "Machine Learning Potentials for atomistic simulations", we noticed he has done previous research on this together with Michele Parrinello in 2007. I thought i t would be a good idea to start with the former paper first to get familiarised with the progress that has been made in the field in the last 9 years and the changes that were introduced.

After giving it a quick skim read, this paper also goes into more depth on the cut-off function for descriptor type methods which is important to understand giving the last paper was not clear enough on that.

Potential Energy Surface (PES) abbreviation used.

Ab initio (from the beginning) methods for finding PES based on DFT are computationally demanding. Therefore empirical approach is becoming more and more popular by the day. Non-ML methods are lenghty and difficult task because you have to **fit parameters of a guessed, physically motivated simple functional** (task we are trying to automate). The database can include **experimental and theoretical data** as well as data from an Ab initio MD run (database used to train model).

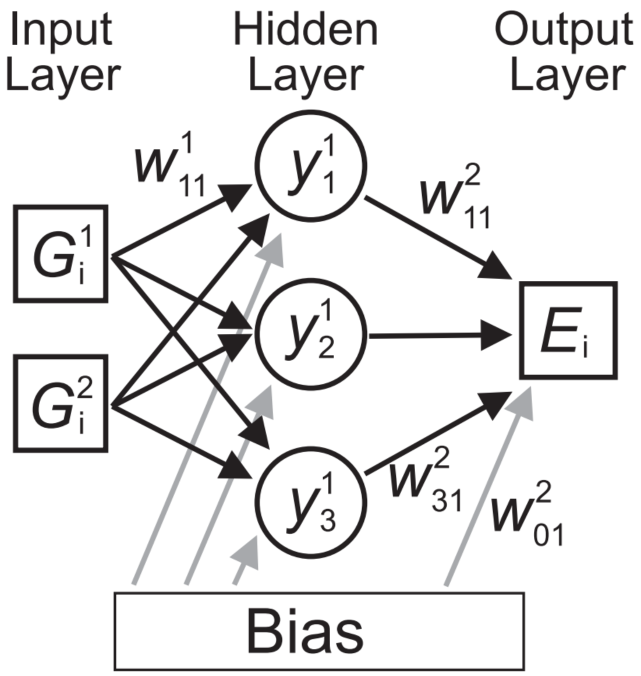
Ab Initio method: https://www.youtube.com/watch?v=awApvINsI0U (https://www.youtube.com/watch?v=awApvINsI0U)

This paper suggests NN based approach using DFT data to finding PES that matches the accuracy of Ab Initio method but at a smaller computing cost. The PES constructed are a function of all coordinates and can be used in systems of arbirtary size.

The paper focuses on finding a potential surface for bulk silicon. Previously it was a challenge to find a potential that works for different phases of the material however this method claims to work in solid semiconducting and liquid metallic phases.

3.1.2 Neural Networks for finding PES using the descriptor method

In simple terms, a supervised learning neural network is a tool that uses a database of (input) points where a certain function is evaluated, and optimises the parameters in the connections to reproduce the input data in a process called training. For this particular application, the training data is obtained from DFT calculations. This produces a general function to which new coordinates can be given to form a specific potential energy surface. A topographical representation of a simple neural network for a simple 2D application is found below.



Inputs are labelled G_i^1 and G_i^2, connections between nodes have "weights" represented as w_{ij}^k which connect nodes i and j in layer k. The nodes in the hidden layer as well as the "output" E_i are called "Bias" nodes as they include an activation function.

From personal experience I know that an activation function is usually a sigmoid or a hyperbolic tan so it is typically non-linear however there are some exceptions.

For the given neural network, the output function E_i can be written as an expression:

$$E_i = f_a^2\left(w_{01}^2 + \sum_{j=1}^{3} w_{j1}^2 f_a^1\left(w_{0j}^1 + \sum_{\mu=1}^{2} w_{\mu j}^1 G_i^\mu\right)\right)$$

All weight parameters are initially chosen randomly. w_{oj}^k is an offset for the activation function as seen in the equation.

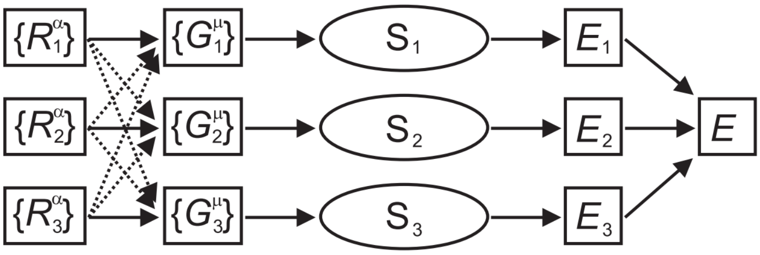
Because at first the weights are random E_i gives the wrong output. Using the DFT calculations, we know the inputs (coordinates) and outputs (energies) for some systems. We can use this data to train our model and find the correct weights. This training can be done by using an error function that needs to be minimised using regression. After the model is trained, we can put in any arbitrary set of coordinates and recover an output.

This is not the final topography of the NN however. The order in which coordinates are fed into the network is important. Another limitation of this simple network is that it cannot be generalised for any system size with an arbitrary degrees of freedom.

Instead, we can represent the total energy as a sum of the contributions from each atom which is an approach used in majority empirical PES models.

$$E = \sum_i E_i$$

This will build upon the topography of the network. Firstly, a pre layer is introduced. $\{R_i^\alpha\}$ is a set of cartesian coordinates α of atom i. These are transformed into a set of symmetry function values $\{G_i^\mu\}$ which describe the local environment of each atom. These inputs $\{G_i^\mu\}$ are then inputted into "subnets" S_i which are identical to the simple network shown before. Then the outputs of the individual subnets are fed into the energy summation equation. This gives the topographical form of:



Each input shown here is one atom and R_i^α represents its coordinates.

The symmetry function G_i^μ must be invariant under transformations. There also needs to be a cutoff function which reduces the effect of atoms to cut down on computing cost just like we saw in the other paper. This function would be f_c and depends on a manually preset parameter R_c

$$f_c(\mathbf{R}_{ij}) \begin{cases} 0.5 \times [cos(\frac{\pi R_{ij}}{R_c}) + 1] & \text{for } \mathbf{R}_{ij} \leq \mathbf{R}_c \\ 0 & \text{for } \mathbf{R}_{ij} > \mathbf{R}_c \end{cases}$$

The radial symmetry function in this paper are used in the shape of sum of gaussians.

$$G_i^1 = \sum_{j \neq 1}^{all} e^{-\eta(R_{ij} - Rs)^2} f_c(\mathbf{R}_{ij})$$

"The summation over all neighbours j ensures the independance of the coordination number" coordination number being the number of atoms immediately surrounding the atom.

Another term in $\{G^\mu\}$ is the angular term which is made for every triplet of atoms in the structure using a the angle in between them θ_{ijk} centrerd at atom i.

$$G_i^2 = 2^{1-\zeta} \sum_{j,k \neq i}^{all} (1 + \lambda \cos\theta_{ijk}) e^{-\eta(R_{ij}^2 + R_{ik}^2 + R_{jk}^2)} f_c(\mathbf{R}_{ij}) f_c(\mathbf{R}_{ik}) f_c(\mathbf{R}_{jk})$$

Here we have the parameters λ, η and ζ which need to be specfied for each system. The three cut off functions make sure a smooth decay that goes to zero at large atomic seperations.

"Several functions of each type with different parameter values are used."

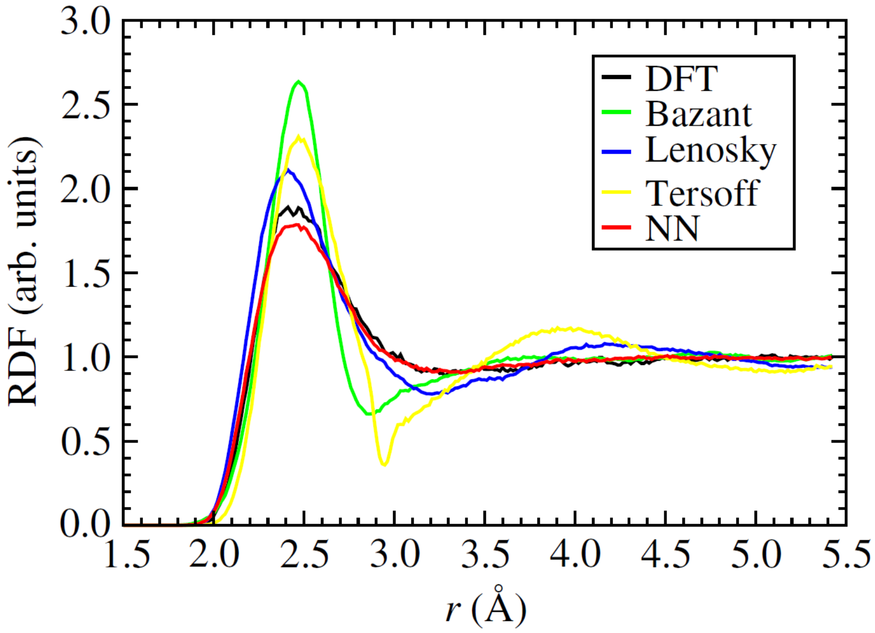
The type or number of symmetry functions is not unique to every system and as many different types can be used as long as the set is suitable for describing the system. These get more complicated for larger systems however the computational cost scales linearly.

3.1.3 Application to bulk silicon (benchmark)

The authors then ran 9000 DFT energy calculations to use for this purpose of which 8200 were used to train the NN and 800 were used to test the NN. Different NN topologies were tested to get a reliable optimisation task. The topology that proved to be the most reliable is one with 2 hidden layers of which each one has around 40 nodes. They also had 48 inputs (symmetry functions) with different parameters η, R_s and ζ.

For normal empirical potentials, it is difficult to describe the correct energetic sequence of empirical potentials while DFT is "in good agreement with experimental data".

To test the NNPs capability of accurately describe disordered structures, they calculated the radial distribution function of a silicon melt at 3000K and here are the results.



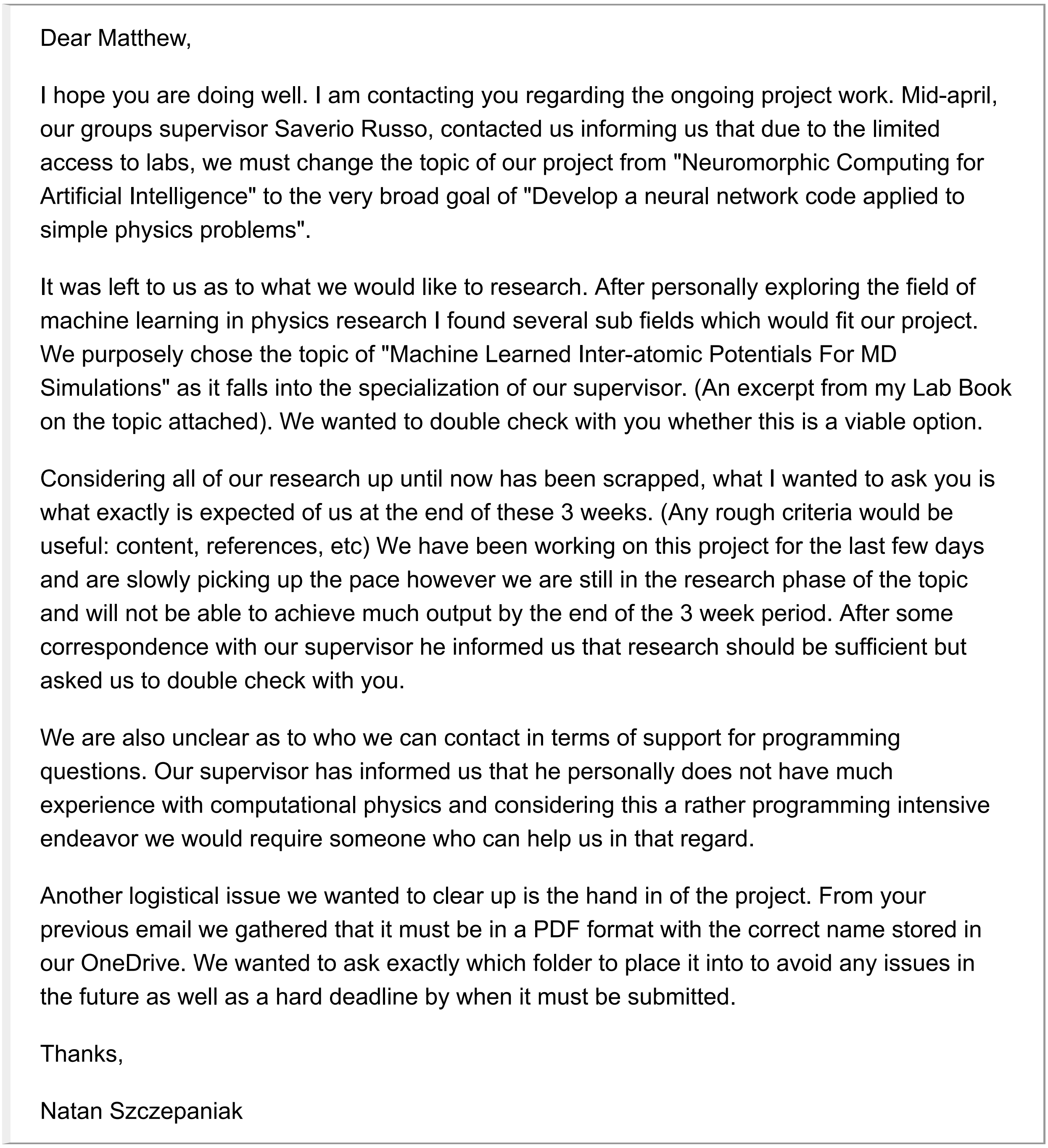
Compared to the classically derived empirical potentials NN method yielded results much closer to the reliable DFT calculatons.

IMPORTANT NOTES TO TAKE AWAY:

- "The accuracy of the NN is limited only by that of training data"
- "For a 64 atom system the NN is currently about 5 orders of magnitude faster than the DFT calculations, and in contrast to DFT the NN scales linearly with system size and is easily parallelized."
- Extension to multicomponent systems requires more intricate symmetry functions. (Basically symmetry functions describe the system.)

3.2 Email to Matthew Bate Regarding Criteria

As we are still unsure of the specific logistics of the project. I decided to Matthew Bate (the MPhys Project Leader) with some questions.



Dear Matthew,

I hope you are doing well. I am contacting you regarding the ongoing project work. Mid-april, our groups supervisor Saverio Russo, contacted us informing us that due to the limited access to labs, we must change the topic of our project from "Neuromorphic Computing for Artificial Intelligence" to the very broad goal of "Develop a neural network code applied to simple physics problems".

It was left to us as to what we would like to research. After personally exploring the field of machine learning in physics research I found several sub fields which would fit our project. We purposely chose the topic of "Machine Learned Inter-atomic Potentials For MD Simulations" as it falls into the specialization of our supervisor. (An excerpt from my Lab Book on the topic attached). We wanted to double check with you whether this is a viable option.

Considering all of our research up until now has been scrapped, what I wanted to ask you is what exactly is expected of us at the end of these 3 weeks. (Any rough criteria would be useful: content, references, etc) We have been working on this project for the last few days and are slowly picking up the pace however we are still in the research phase of the topic and will not be able to achieve much output by the end of the 3 week period. After some correspondence with our supervisor he informed us that research should be sufficient but asked us to double check with you.

We are also unclear as to who we can contact in terms of support for programming questions. Our supervisor has informed us that he personally does not have much experience with computational physics and considering this a rather programming intensive endeavor we would require someone who can help us in that regard.

Another logistical issue we wanted to clear up is the hand in of the project. From your previous email we gathered that it must be in a PDF format with the correct name stored in our OneDrive. We wanted to ask exactly which folder to place it into to avoid any issues in the future as well as a hard deadline by when it must be submitted.

Thanks,

Natan Szczepaniak

<u>**Day 4 - 29/05/20**</u>

4.1 [3] Machine Learning Potentials for atomistic simulations

Skim Read Discussion

This paper seems to be just an iteration of the 2007 paper by Behler as it covers a very similar concept to the last one however, it describes different descriptors that can be used. It also describes the progress that has been made in the field in the time and all of the other approaches which were developed in that time. It describes different methods and approaches of generating a PES in the form of different descriptors. For now however, to keep things simple I will only discuss the atom centered symmetry functions as described in the previous paper as it seems to be the flagship method in the field.

> *In many cases, the availability of sufficiently efficient interatomic potentials providing reliable energies and forces has become a serious bottleneck for performing these simulations*

Paper discusses the "applicability and limitations" of ML potentials using different descriptors for the system.

4.1.1 Introduction

ML was used in Condensed matter physics for analysis however recently it has been introduced in actual modelling. When this method was first introduced (1995) it was met with a lot of scepticism. Due to the amount of recent progress done in ML it has grown into a mature set of methods used among many disciplines. Nowadays, using ML methods to derive empirical interatomic potentials has become a new trend to cut on computational cost significantly. These methods are being used in fields like drug design and condensed matter resarch.

ML definition by Tom M. Mitchell:

> *A computer program is said to learn from experience E with respect to some class of tasks T and performance measure P, if its performance at tasks in T, as measured by P, improves with the experience E.*

Applying ML to experimental data analysis is not a new task. Using ML to analyse theoretical data that is difficult to analyse manually is. Using it this way, we can discover unknown relationships and new materials. This project however focuses on representing interatomic potentials accurately at a minimum computational cost.

4.1.2 Analytical Potential Energy Surface function (PES)

> *Multidimensional real-valued function providing the potential energy of a system as a function of atomic positions.*

If atomic positions, nuclear charge and the total charge are known the entire system can be described by the Electronic Hamiltonian. The concept of PES is based on the Born-Oppenheimer approximation which seperates the dynamics of the nuclei and electrons.

Molecular dynamics requires energies and forces for a very large amount of atomic configurations.

These can be either calculated "on the fly" which requires you to calculate energies and forces which is accurate however quite inefficient. Methods like DFT based ab initio are extremely accurate however they are limited to a few hundred atoms and "simulation times significantly shorter than 1ns".

Analytic empirical potentials are an approach to make this process more efficient. However, at the cost of this increased efficiency comes a downside of reduced accuracy. What the field of ML Potentials introduces is the efficiency of empirical potentials with the accuracy of the (first principles) DFT ab initio process. This is a very quickly growing field according to this paper. I have also seen quite a lot about this on the internet.

So the approach we will be taking for this project will be similar to previous methods of deriving emprirical potentials but with the use of ML to reach better accuracy.

> *The information required to evaluate potential should not include any classification of atoms into types beyond the specification of nuclear charge.*

This allows the chemical environment to change over time. Ideally these potentials should be a general form not specified for certain atomic interactions which just shows that ubiased ML methods are ideal for this task.

The requirements of an ML potentials are outlined in the paper as:

- employs a ML method to construct a direct functional relation between the atomic configuration and its energy;
- does not contain any physical approximations apart from the chosen reference electronic structure method used in its construction;
- is developed using a consistent set of electronic structure data.

For ML potentials to be created, the raw coordinates of atoms need to be transferred into a format usable by the ML package. This is done with descriptors (More described in the later sections that I will discuss next week). After the coordinates are converted into descriptor functions, these are fed into the ML algorithm for optimisation which generates the energies and required forces. This then is passed onto a simulation software such as LAMMPS to run the MD simulation.

(I will follow the discussion of this paper next week.)

4.2 Logistics of Project

Unfortunately Matthew Bate has not gotten back with criteria yet. I will be uploading this lab book on onedrive under the correct name regardless. There are multiple ways of exporting the file as PDF I explored different ones including Download->"as .HTML" and then formatting it to keep the same formatting.

Next week I will focus on one method that we agreed upon with our collegues and try to get someones implementation working to demonstrate the process.