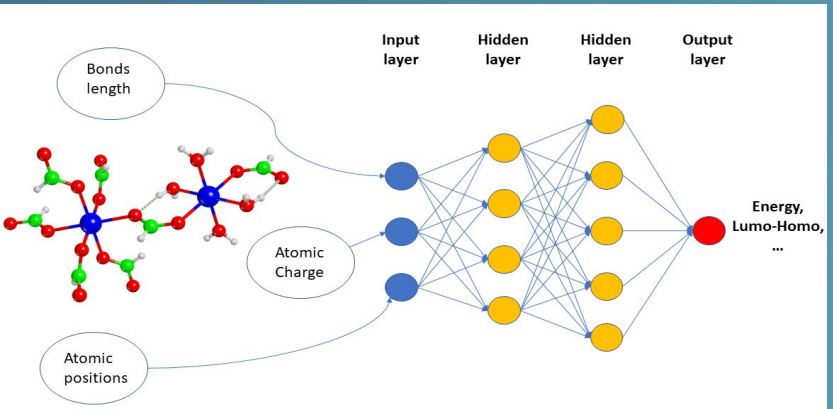
Neural Networks in Quantum Chemistry

By Scott le Roux and Joseph Sleiman



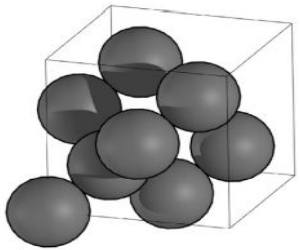
TensorFlow

python

Quantum Chemistry

- Study of molecular interactions on the atomic and subatomic scale
- Inherently quantum in nature
- Density Functional Theory (DFT) has led the field in solving quantum chemical problems until recently

Diamond Cell



Coulomb matrix

36.9	23.3	14.3	9.3	14.3	9.3	14.3	9.3
23.3	36.9	23.3	14.3	23.3	14.3	23.3	14.3
14.3	23.3	36.9	23.3	14.3	12.2	14.3	12.2
9.3	14.3	23.3	36.9	12.2	14.3	12.2	14.3
14.3	23.3	14.3	12.2	36.9	23.3	14.3	12.2
9.3	14.3	12.2	14.3	23.3	36.9	12.2	14.3
14.3	23.3	14.3	12.2	14.3	12.2	36.9	23.3
9.3	14.3	12.2	14.3	12.2	14.3	23.3	36.9

Coulomb Matrix Example [1]. Diagonal entries can be seen as the interaction between an atom and itself. Off-Diagonal entries are the Coulombic repulsion between two atoms

$$M_{ij}^{\text{Coulomb}} = \begin{cases} 0.5Z_i^{2.4} & \text{for } i = j \\ \frac{Z_i Z_j}{R_{ij}} & \text{for } i \neq j \end{cases}$$

DScribe

- Python based library transforming molecular data into a form readable by Machine Learning models

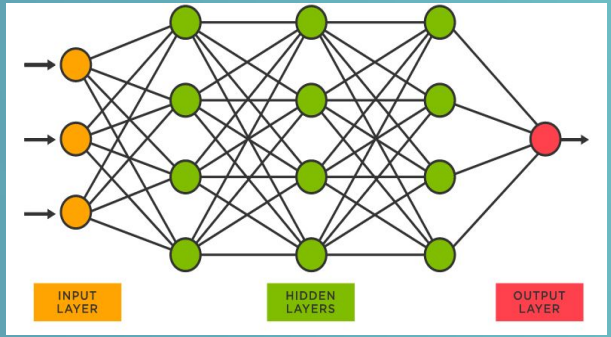
Coulomb Matrix (CM)

Many-Body Tensor Representation (MBTR)

Atom-Centered Symmetry Functions (ACSF)

Smooth Overlap of Atomic Positions (SOAP)

Neural Networks

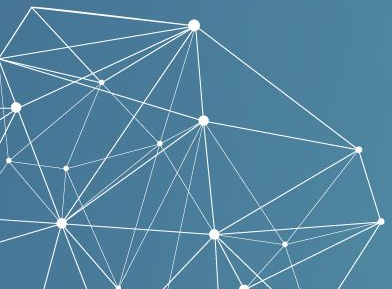


- Feed-Forward Neural Networks:
  - Descriptor fed into input layer
  - Hidden layers and non-linear activation functions
  - Single-neuron output layer predicting final internal energy of molecule
- Neural Networks use numerical optimisation to determine structure-property relationships of molecules
- TensorFlow - deep learning Python library to implement the Neural Networks
- KerasTuner - software used to optimise the hyperparameters of the Neural Network in order to minimise the MSE

QM9 dataset

- 134,000 organic molecules with energetic and thermodynamic properties
- Industry standard dataset (available at http://quantum-machine.org/datasets/)
- Allows for Supervised learning

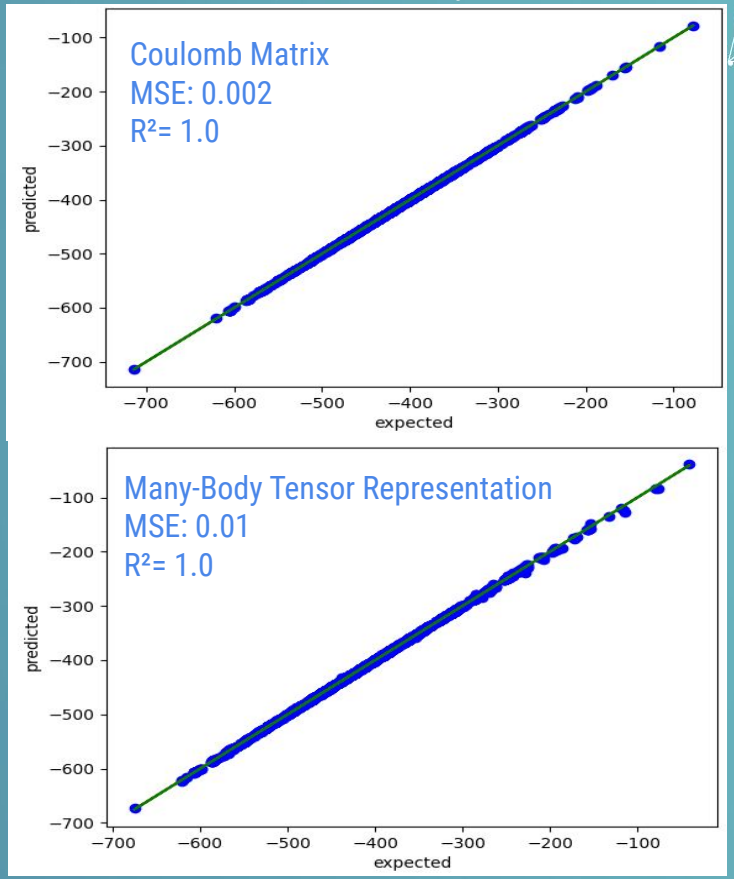
Model Optimization and Results



Coulomb Matrix & Many-Body Tensor Representation

Hyperparameters	CM	MBTR
Learning Rate	.0001	.001
Batch Size	32	32
Optimizer	ADAM	ADAM
Hidden Layer Network Topology	[256]	[64,64,32,128]
Activation	softplus	ELU
Weight Initializer	glorot	glorot
MSE	.002	.01

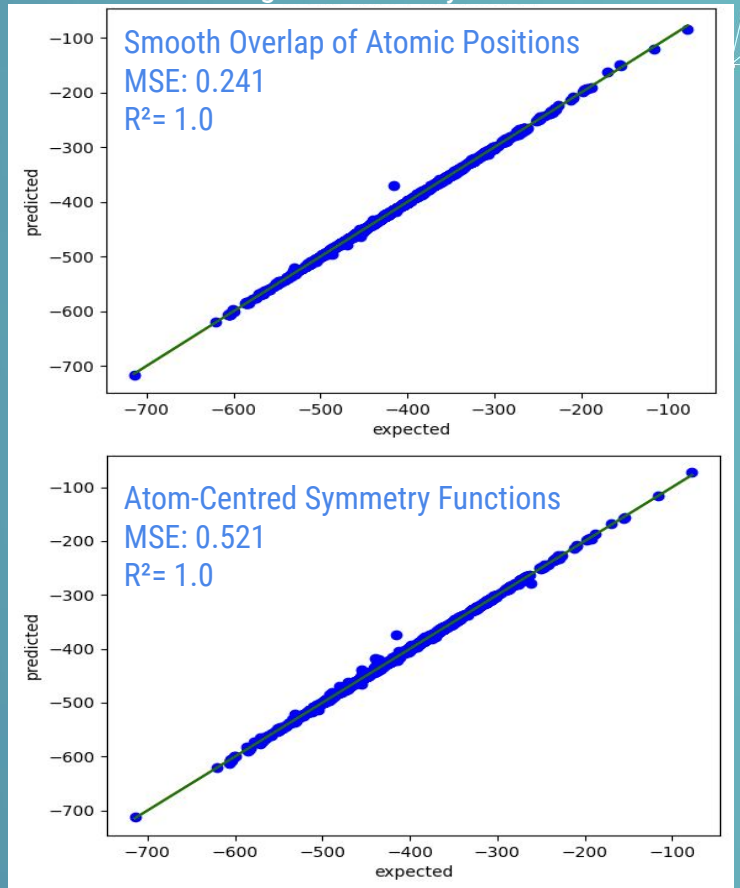
Linear Regression Analysis of Test Set



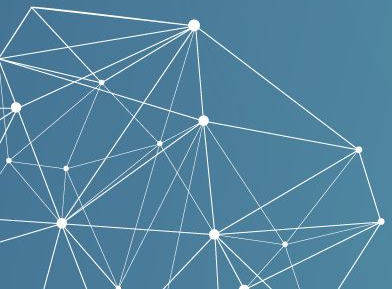
Smooth Overlap of Atomic Positions & Atom Centered Symmetry Functions

Hyperparameters	SOAP	ACSF
Learning Rate	.001	.0001
Batch Size	32	32
Optimizer	ADAM	ADAM
Hidden Layer Network Topology	[256,32,128]	[128,32,16]
Activation	ELU	softplus
Weight Initializer	He Uniform	He Uniform
MSE	.241	.521

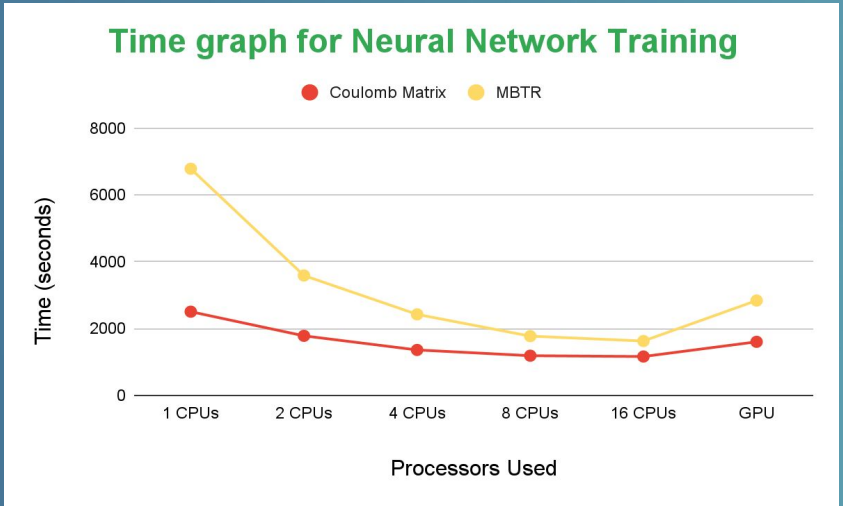
Linear Regression Analysis of Test Set

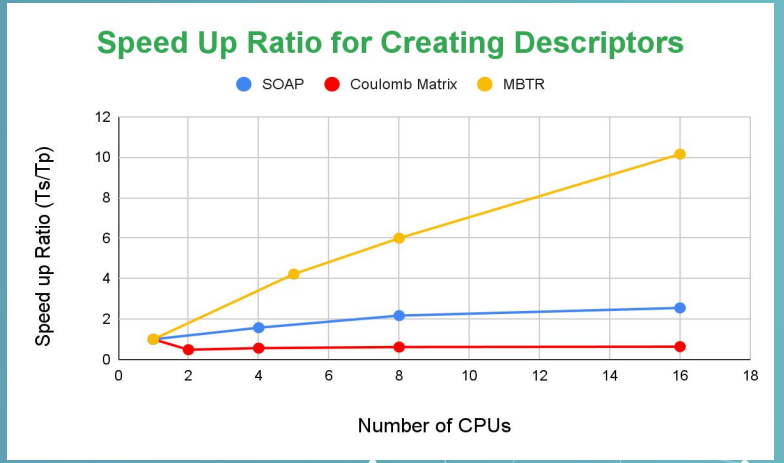


GPU vs CPU

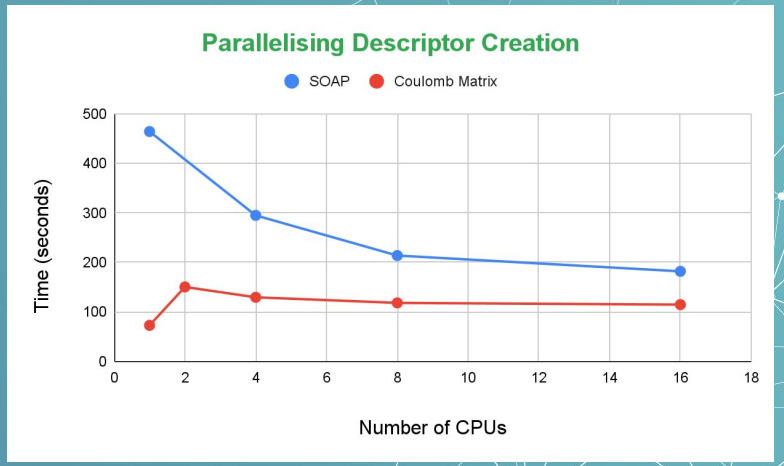


Comparison of Computation Times





- For Training Neural Networks GPU is faster than CPU but parallelised CPUs are optimal
- For more complex descriptors like SOAP, the more cores used yields better results.
- For simple descriptors like the Coulomb Matrix, using 1 core is fastest due to parallel overheads.



Conclusions and Future Work

- The Coulomb Matrix representation produced the best results in terms of *energy* prediction (MSE: ~0.002) and was the most computationally efficient descriptor
- GPU utilisation provides a significant speedup compared to single core CPU computations
- Parallelisation of multiple CPU cores provides an even faster speedup than GPUs in most cases
- The best Neural Network topology varied depending on the descriptor
- One can use Neural Networks to predict other molecular properties such as charges or spin using a multi-neuron output layer and *local* descriptors such as ACSF