The Theory of Machine Learned-Potentials

Contents

1 Introduction

Machine-learned potentials (MLPs) are represented by generic functions containing several dozens or even hundreds of parameters, i.e., very often a much larger number of parameters than typically defined for most traditional forcefields. The limited number of descriptors in most traditional forcefields constrains the degree to which they can match a large training set of material configurations. The smaller the number of atomic potential parameters, the more computationally efficient it tends to be, but the smaller the configuration space to which a potential is trained, the more limited is its applicability to material configurations outside of its training set. The larger parameter spaces represented by MLPs enables them to describe a larger material configuration space with improved and relatively uniform fidelity, thus maximizing the value of large training sets such as can be generated with *ab initio* computational methods. This advantage comes at a cost. An MLP described by a large parameter set requires correspondingly larger computational efforts both to develop and to use it. Furthermore, the greater versatility of MLPs, and a functional form amenable to automated optimization methods, requires that one abandons older functional forms in which the individual terms in the potential claim independent physical meaning such as Coulomb and Van der Waals interactions. Only the entire MLP taken as an integral whole has physical meaning. This fact implies that an MLP optimized on a particular training set of material configurations is different from any other MLP optimized on any different training set or optimized with different fitting criteria. Adding a new structural configuration, or a new alloy constituent, or changing the fitting criteria, requires an entirely new fit.

The overall workflow of most MLP approaches consists of two, sometimes iterative steps, namely,

- The generation of a large dataset of atomic structures and the associated energies, forces, stress tensors, and possibly other properties as arising from DFT calculations.
- The generation of the MLP in terms of an efficient and appropriate set of atomic structure descriptors by determining the open parameters of that description through fitting to the DFT data. This second step is best captured by the formula $Y = f(X;\omega)$, where X denotes the set of atomic descriptors of the atomic structures, Y is the set of properties to be matched by the fitting process, and ω represents additional parameters increasing the flexibility of the approach.

Knowledge from *ab initio* calculations is reduced in machine-learned potentials to the most fundamental structure-property relationship. To make this information accessible to machine learning a unique, flexible, and efficient description of the atomic arrangement in the training set structures is needed. This task is accomplished by dividing each structure into local spherical clusters centered at a given atom and specified by a cutoff radius.

Beyond this overarching approach of using local atomic clusters, different schemes have emerged to represent the resulting local geometries following the requirements of invariance of the representation under translations, rotations, and permutations.

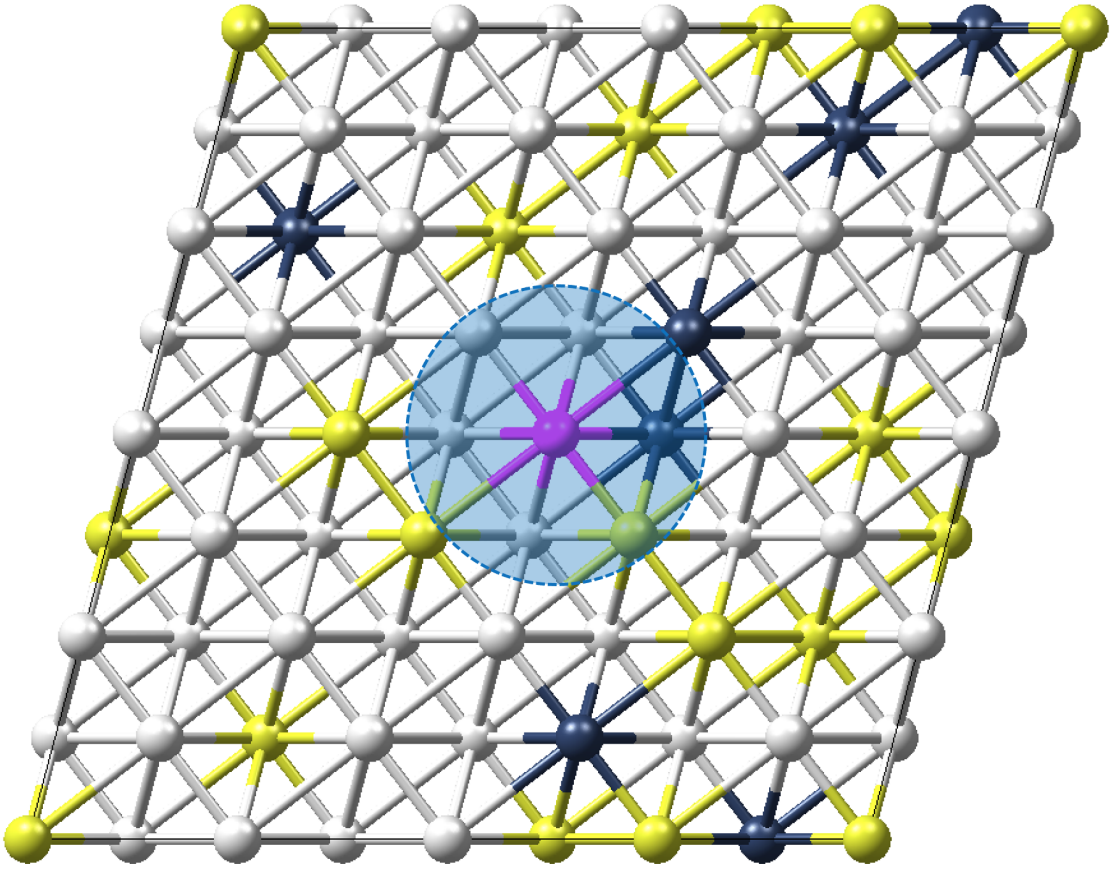

Figure1: Illustration of the environment-dependence of atomic energies for a system with periodic boundary conditions. The energy contribution of the atom in pink depends on the positions of all the atoms within the dashed sphere that is defined by the cutoff radius Rc of the symmetry functions.

2 Neural Network Potentials

In the neural network schemes these symmetry requirements are satisfied by atom-centered symmetry functions [7]. To ensure that the radial terms of these symmetry functions are zero for all radii r larger than the cutoff radius r_C a cutoff function such as,

$$f_C\left(r, r_C\right) = \begin{cases} \frac{1}{2}\left[\cos\left(\frac{\pi r}{r_C}\right) + 1\right] & \text{for } r \leq r_C \\ 0 & \text{for } r > r_c \end{cases} \tag{1}$$

or

$$f_C\left(r, r_C\right) = \begin{cases} \tanh^3\left(1 - \frac{r}{r_C}\right) & \text{for } r \leq r_C \\ 0 & \text{for } r > r_c \end{cases} \tag{2}$$

is introduced [7], [8]. With this cutoff function the radial symmetry function G_i^{rad} centered on atom i is defined by the sum over all its neighbors j within the cutoff radius r_C

$$G_i^{\mathrm{rad}}\left(r_C, r_s\right) = \sum_{j \neq i} f^{\mathrm{rad}}\left(r_{ij},\ r_s\right) f_C\left(r_{ij},\ r_c\right), \tag{3}$$

with the radial Gaussian function

[7] J. Behler, and M. Parrinello, *Generalized Neural-Network Representation of High Dimensional Potential-Energy Surfaces*, Phys. Rev. Lett. **98**, 146401 (2007)

[8] M. P. Bircher, A. Singraber, and C. Dellago, *Improved Description of Atomic Environments using Low-cost Polynomial Functions with Compact Support*, Mach. learn.: sci. technol. **10**, 2632-2153 (2021)

$$f^{\mathrm{rad}}\left(r, r_s\right) = \; e^{-\eta\left(r-r_s\right)^2} \; . \tag{4}$$

The width of the Gaussian is defined by η and can be shifted by the radial distance r_s. Angular dependencies can be considered by introducing the angular function

$$f^{\mathrm{ang}}(\vartheta, \lambda, \; \zeta) = \left(1 + \lambda \cos\left(\vartheta\right)\right)^{\zeta} \quad , \tag{5}$$

where the angle ϑ is between the lines joining the central atom to any other two atoms within the cutoff radius r_C of the central atom. The maximum and minimum of this angular function lie at 0 and π for $\lambda = 1$. This is the opposite for $\lambda = -1$. The decay of the angular function with ϑ increases with increasing ζ. To be meaningful ζ must be larger than 1. With the angular function defined as in Eq. (5), the narrow angular symmetry function centered on atom i is defined by

$$\begin{aligned} G_i^{\mathrm{ang},n} \; \left(r_C, r_s, \lambda, \zeta\right) = \; 2^{(1-\zeta)} \sum_{j \neq i, k > j} & f^{\mathrm{rad}}\left(r_{ij}, r_s\right) f^{\mathrm{rad}}\left(r_{ik}, r_s\right) f^{\mathrm{rad}}\left(r_{jk}, r_s\right) \\ & f^{\mathrm{ang}}\left(\, \vartheta_{ijk}, \lambda, \zeta\right) f_C\left(r_{ij}, r_C\right) f_C\left(r_{ik}, r_C\right) f_C\left(r_{jk}, r_C\right) \quad . \end{aligned} \tag{6}$$

This narrow angular symmetry function considers the interatomic distances r_{jk} between atoms j and k that span the angle with the central atom i. In contrast the wide angular form,

$$\begin{aligned} G_i^{\mathrm{ang},w} \; \left(r_C, r_s, \lambda, \zeta\right) = \; 2^{(1-\zeta)} \sum_{j \neq i, k > j} & f^{rad}\left(r_{ij}, r_s\right) f^{\mathrm{rad}}\left(r_{ik}, r_s\right) \\ & f^{\mathrm{ang}}\left(\, \vartheta_{ijk}, \lambda, \zeta\right) f_C\left(r_{ij}, r_C\right) f_C\left(r_{ik}, r_C\right) \quad , \end{aligned} \tag{7}$$

does not consider this distance.

Finding a suitable set of symmetry functions is vital for correctly describing the local environments around each atom. For this purpose, radial symmetry functions are generated on a grid by using the method proposed by Imbalzano *et al.* [9]. For centered radial symmetry functions, where the radial shift r_s is zero, the width of the Gaussians varies with

$$\eta_m = \left(\frac{n^{\frac{m}{n_r}}}{r_C} \right)^2 \quad , \tag{8}$$

where n_r defines the number of intervals chosen and $m = 0, 1, \ldots, n_r$. Alternatively, a set of n_r shifted radial symmetry functions can be defined with the radial shift,

$$r_{s,m} \; = \frac{r_c}{n_r^{\left(\frac{m}{n_r}\right)}} \; , \tag{9}$$

and the Gaussian widths

$$\eta_{s,m} \; = \frac{1}{\left(r_{s,n_r-m} \; - r_{s,n_r-m-1}\right)^2} \; . \tag{10}$$

Thus, the generated symmetry functions are narrow close to the central atom and become wider with increasing distance from the center (see figure below).

Such a grid based spacing can also be applied to the radial terms of the angular symmetry functions using the approach by Gastegger *et al.* [10]. In this approach the distance Δr between the points on a grid from r_0 to r_n consisting of n_a points is obtained with

$$\Delta \, r \; = \frac{r_n - r_0}{n_a - 1} \quad . \tag{11}$$

In the case of centered Gaussian functions r_{low} is set to the lower bound, r_n to $(r_C - 0.5)$ and all shift radii are set to zero. The widths of the Gaussians are defined by

$$\eta_i \; = \frac{1}{2 \; \left(r_{\mathrm{low}} + \; i \; \Delta \, r\right)^2} \; . \tag{12}$$

Shifted angular symmetry functions have the radial terms shifted by

[9] G. Imbalzano, A. Anelli, D. Giofré, S. Klees, J. Behler, and M. Ceriotti, J. Chem. Phys. **148**, 241730 (2018)
[10] M. Gastegger, L. Schwiedrzik, M. Bittermann, F. Berzsenyi, and P. Marquetand, J. Chem. Phys. **148**, 241709 (2018)

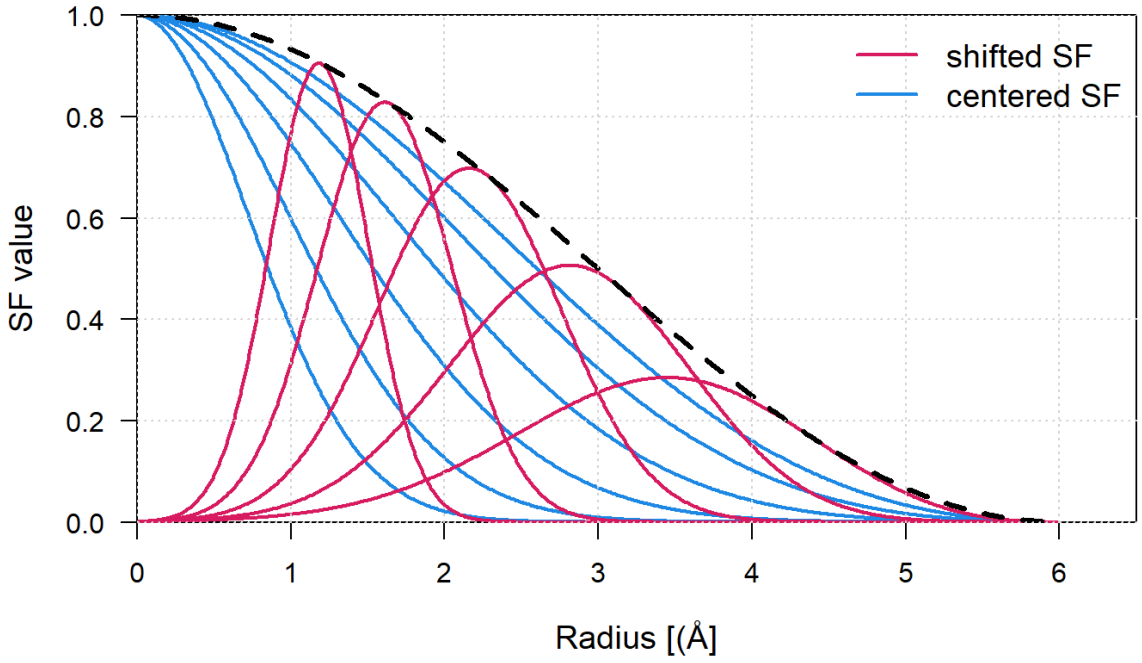

Figure2: Example of radial symmetry functions generated with $n_r = 5$ and $r_c = 6$ Å using the scheme of Imbalzano *et al.* [9]. The blue curves show the centered symmetry functions, i.e., with $r_s = 0$, and η varying as described in Eq. (8). The red curves show shifted symmetry functions with r_s as described by Eq. (9). and η as described by Eq. (10). The black dashed line indicates the cutoff function for $r_c = 6$ Å.

$$r_{s,i} \;=\; r_{\text{low}} + i\,\Delta\, r \tag{13}$$

and the widths described by

$$\eta = \frac{1}{2\Delta\, r^2} \quad . \tag{14}$$

This structured approach allows for a straightforward and reliable definition of the symmetry functions depending only on a set of parameters.

Once the geometry of the atomic neighborhood within each local cluster has been uniquely specified using the symmetry function descriptors of Equations (3), (6) and (7), development of the MLP proceeds by representing the total energies in terms of these descriptors. In the neural-network approach proposed by Behler and coworkers the total energy is given by Eq. (15) [7], [1], [2], [3], [4], [5], [6] . The rationale behind this equation is sketched below which illustrates the use of Equation (15) in terms of a procedure applied to the symmetry functions, i.e., the local descriptors G_i, moving from left to the right in the figure. At each level, a linear combination of the input quantities (the local descriptors G_i are used at the initial level) is first calculated using parameters a and b. Then a non-linear activation function f is applied to this linear combination. Common choices for activation functions are sigmoid functions, hyperbolic tangents, Gaussians, or simple linear functions. This procedure is convenient for recursion yielding a total energy value. If this value agrees

[1] J. Behler, *First Principles Neural Network Potentials for Reactive Simulations of Large molecules and Condensed Systems*, Angew. Chem. Int. Ed. **56**, 12828 (2017); Angew. Chem. *129*, 13006 (2017)
[2] J. Behler, *Atom-centered symmetry functions for constructing high-dimensional neural networks*, J. Chem. Phys. **134**, 074106 (2011).
[3] J. Behler, *Neural network potential-energy surfaces in chemistry: a tool for large-scale simulations*, Phys. Chem. Chem. Phys. **13**, 17930 (2011).
[4] J. Behler, *Representing potential energy surfaces by high-dimensional neural network potentials*, J. Phys.: Condens. Matter **26**, 183001 (2014)
[5] J. Behler, *Constructing High-Dimensional Neural Network Potentials: A Tutorial Review*, Int. J. Quant. Chem. **115**, 1032 (2015)
[6] J. Behler, *Perspective: Machine learning potentials for atomistic simulations*, J. Chem. Phys. **145**, 170901 (2016), J. Chem. Phys. **145**, 219901 (2016)

with the *ab initio* result for the training set to within the user-selected tolerance, the procedure is considered converged. Otherwise, it is repeated with improved guesses for the parameters a and b. These improved guesses can be generated from simulated annealing, genetic algorithms, Levenberg-Marquardt, multi-stream Kalman filters [11] or related optimization algorithms. The robustness of neural network methods is widely documented, having been applied in diverse contexts to a variety of complex problems such as speech and pattern recognition. Neural networks can be used to approximate any kind continuous function [13], [14], [15] and are thus highly adaptable to reproduce complex energy hypersurfaces. With the specification of the local descriptors G by Equations (3), (6), and (7) and their use in Eq. (15) where the energy for atom i is calculated, the power of neural networks becomes accessible to atomistic simulations.

$$E_i = f\left(b_1^3 + \sum_{k=1}^{3} a_{k1}^{23} f\left(b_k^2 + \sum_{j=1}^{3} a_{jk}^{12} f\left(b_j^1 + \sum_{i=1}^{2} a_{ij}^{01} G_i\right)\right)\right) \tag{15}$$

The total energy is simply the sum of the atomic energies Eq. (15) over all N^{at} atoms,

$$E^{\mathrm{pot}} = \sum_{i}^{N^{\mathrm{at}}} E_i \; . \tag{16}$$

In turn the atomic forces F_i are obtained by applying the chain rule [11] ,

$$\vec{F}_i = -\vec{\nabla}_i E = -\sum_{j=1}^{N^{\mathrm{at}}} \vec{\nabla}_i E_j = -\sum_{j=1}^{N^{\mathrm{at}}} \sum_{k=1}^{N_j^{\mathrm{SF}}} \frac{\partial E_j}{\partial G_{j,k}} \vec{\nabla}_i G_{j,k}, \tag{17}$$

with N_j^{SF} for the number of symmetry functions used to describe atom j.

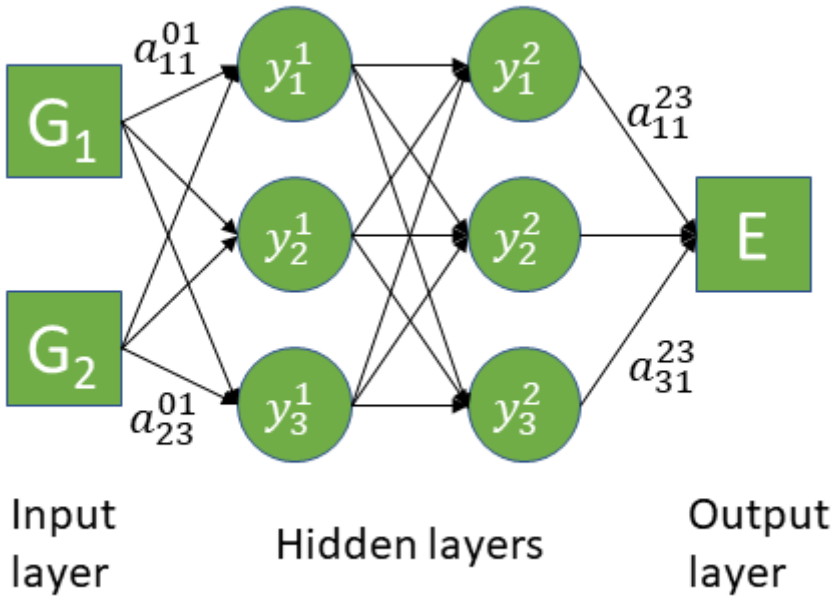

Figure3: Schematic structure of a feed-forward neural network defining the functional relationship between a two-dimensional input vector $G = (G_1, G_2)$ that describes the atomic configuration and the potential energy E.

Neural network potentials are generated in *MedeA* by the *Machine-Learned Potential Generator* using the n2p2 code by Singraber *et al.* [11] with the weights and biases of the neural network composed of two hidden

[11] A. Singraber, T. Morawietz, J. Behler, and C. Dellago, *Parallel Multistream Training of High-Dimensional Neural Network Potentials*, J. Chem. Theory Comput. **15**, 3075-3092 (2019)
[13] K. Hornik, Neural Netw. **4**, 251-257 (1991)
[14] M. Leshno, V. Y. Lin, A. Pinkus, and S. Schocken, Neural Netw. **6**, 861-867 (1993)
[15] A. Kratsios, The Universal Approximation Property, Ann. Math. Artif. Intell. (2021).

layers, each containing a user definable number of neurons, optimized utilizing a Kalman filter fading memory. The thus generated NNP can then be used in *MedeA* with the n2p2 LAMMPS interface [12].

3 Spectral Neighbor Analysis Potentials

An alternative scheme put forward by Bartok and coworkers expresses the atomic arrangement within each local cluster in terms of the density of the atoms as given by Eq. (18) [16], [17]. In this approach, the δ-distribution selects the neighboring sites within the local environment. (Within the soft overlap of atomic positions (SOAP) approach, each δ-distribution is replaced by a Gaussian distribution to make the density a smoothly varying function). The density could then be expanded in a spherical harmonic basis set as sketched in Eq. (19) [1,2]. However, a direct expansion in spherical harmonics would not be invariant under rotations. This problem is solved by introducing a "bispectrum" of spherical harmonics, which is rotationally invariant but nevertheless includes all the angular information contained in the material configurations being described [16] [17] . In addition, the radial dependence of the atomic density within each cluster is encoded in terms of a fictitious "angle" θ_0, which enables one to represent the atomic density as an expansion over the mathematically complete set of four-dimensional spherical harmonics, with θ and ϕ being the usual angles, and θ_0 holding the radial information scaled to the cutoff radius divided by π. For a given atom distribution within a local cluster, this four-dimensional representation of the density is thus fully specified by the coefficients $u^j_{m,m'}$ entering Eq. (19). Finally, this representation may be transformed to one in terms of the bispectrum coefficients B as given by Eq. (20), which carry all the information about atomic distributions necessary for constructing MLPs. In Eq. (20), the C 's are the familiar Clebsch-Gordan coefficients representing the integrals of products of three-dimensional spherical harmonics with their complex conjugates. In passing it should be noted that Eqs. (3), (6), (7), and (20) are alternative but equivalent complete specifications of the structures entering the training set that depend only on the relative coordinates within each atomic cluster. As such, both representations are invariant under translations, rotations, and permutations.

$$\rho_i(r) = \delta(r) + \sum_{r_{ij}<r_c} f_c(r_{ij}) \cdot w_j^{\text{atom}} \cdot \delta(r - r_{ij}) \tag{18}$$

$$\rho_i(r) = \sum_{j=0}^{\infty} \sum_{m,m'=-j}^{j} u^j_{m,m'} U^j_{m,m'}(\theta, \phi, \theta_0) \ . \tag{19}$$

$$B_{j_1,j_2,j} = \sum_{m_1,m_1'=-j_1}^{j_1} \sum_{m_2,m_2'=-j_2}^{j_2} \sum_{m,m'=-j}^{j} \left(u^j_{m,m'}\right) C^{jm}_{j_1 m_1 j_2 m_2} \times u^{jm'}_{j_1 m_1', j_2 m_2'} u^{j_1}_{m_1',m_1} u^{j_2}_{m_2',m_2} \tag{20}$$

Once the bispectrum coefficients are determined, the total energies and forces calculated for the structures of the training set are written as a sum over atoms in terms of these coefficients as given by Eq. (20) [16] [17]. Again, starting from initial guesses, the parameters β are then varied until the total energy determined from these values agrees with the value arising from the corresponding *ab initio* calculation to a specified accuracy. Eventually, this procedure leads to the Gaussian Approximation Potential (GAP), which was originally proposed by Bartók and coworkers and uses Bayesian regression to determine the parameter values [16].

$$E_{\text{SNAP}} = \sum_{i=1}^{N} \beta_0^{\alpha_i} + \sum_{i=1}^{N} \sum_{k=\{j,j_1,j_2\}} \beta_k^{\alpha_i} B_k^i \tag{21}$$

$$F^j_{\text{SNAP}} = -\sum_{i=1}^{N} \beta^{\alpha_i} \frac{\partial B^i}{\partial r_j}$$

In contrast, the SNAP potential as introduced by Thompson *et al.* [17], [19], while using the same representation of the local neighborhoods as GAP, seeks to determine the parameters β from linear regression rather

[12] A. Singraber, J. Behler and C. Dellago, *Library-Based LAMMPS Implementation of High-Dimensional Neural Network Potentials*, J. Chem. Theory Comput. **15**, 1827-1840 (2019)
[16] A. P. Bartók, M. C. Payne, R. Kondor and G. Csányi, *Gaussian Approximation Potentials: The Accuracy of Quantum Mechanics, without the Electrons*, Phys. Rev. Lett. **104**, 136403 (2010)
[17] A. P. Thompson, L. P. Swiler, C. R. Trott, S. M. Foiles and G. J. Tucker, *Spectral neighbor analysis method for automated generation of quantum-accurate interatomic potentials*, J. Comput. Phys. **285** 316-330 (2015)
[19] A. Thompson, *SNAP: Spectral neighbor analysis method for automated generation of quantum-accurate interatomic potentials for LAMMPS*, talk given at the LAMMPS Users' Workshop and Symposium, August 2015

than Gaussian process regression. This choice offers distinct advantages. In particular, according to the authors this step makes the fitting process more robust and it decouples the speed of molecular dynamics from the size of the training set thereby facilitating application of the MLP. Linear regression also accommodates larger training sets, and finally, it enables straightforward sensitivity analysis.

In practice, the generation of SNAP potentials proceeds along the following path:

1. A training set of structures is built and used to compute total energies, forces, and stress tensors from ab initio calculations.
2. The local environment of each atom in each structure is represented via the 4D spherical harmonics expansions in terms of the bispectrum coefficients B, which take the role of local structural descriptors.
3. Linear regression is used to obtain the coefficient values β entering the expansion of the total energy in the local structural descriptors B.
4. Finally, hyper-parameters such as the number of structure descriptors, cutoff radii of the local atomic clusters, and weights of the atomic types are optimized using a numerical package such as the DAKOTA software or any other optimization code such as the NEWUOA code by Powell [20], which adopted in the present context.

A similar approach is used within the *Machine-Learned Potential Generator* implementation in *MedeA*, which likewise employs the FitSNAP code, but uses *MedeA VASP* for the training set calculations and Powell's Derivative-Free Optimization solvers for the optimization of the hyper-parameter values in an outer loop, namely, the relative cutoff radii and weights for the atom types.

[20] M. J. D. Powell, *The NEWUOA software for unconstrained optimization without derivatives.* In: G. Di Pillo and M. Roma (eds), *Large-Scale Nonlinear Optimization. Nonconvex Optimization and Its Applications* **83** (Springer, Boston 2006)