Machine Learning for Potential Energy Surface Approximation

Khachik Sargsyan

Sandia National Laboratories, Livermore, CA

Data Science for Chemistry Reading Group
SNL-CA
October 23, 2018

Overview

- Reading group goals
- Machine learning
  - Generalities
  - Relevance to DOE
- **Constructing High-Dimensional Neural Network Potentials: A Tutorial Review**, by Jörg Behler, *International Journal of Quantum Chemistry*, 115, pp. 1032-1050, 2015.
- Potential energy surface (PES)
- ML-PES ingredients
  - Input representation
  - Functional form
  - Loss function
  - Training
- Summary and literature

Reading Group Goals

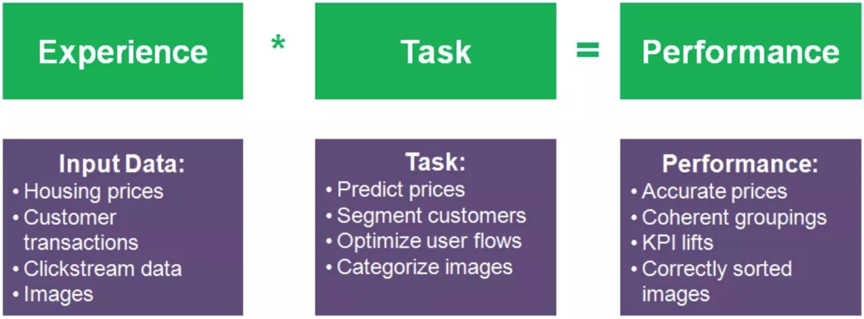
- DS/ML success of last decade
  - For our purposes Data Science = Machine Learning
- Recognized challenges, particularly for physical systems
- Growing body of literature on chemistry applications
  - Clear potential for significant impact
- Forum for open discussion of existing work and potential new directions
- Foster collaboration [and future proposals?]
- Bi-weekly talks. Pick a paper and volunteer!
- Meant to be interactive
  - I am not an expert in today's topic

Machine Learning

Mitchell’1997 definition

[http://www.cs.cmu.edu/∼tom/mlbook.html]

Learning from experience for a given task.



Experience * **Task** = **Performance**

Input Data:

- Housing prices
- Customer transactions
- Clickstream data
- Images

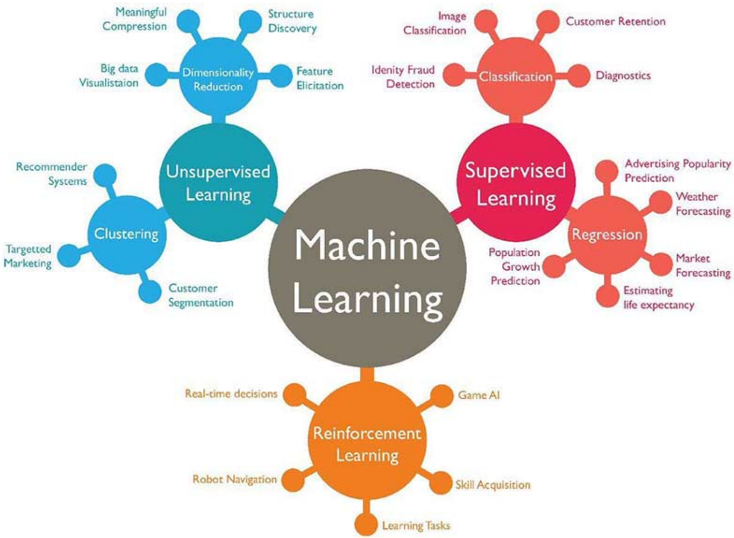
Task:

- Predict prices
- Segment customers
- Optimize user flows
- Categorize images

Performance:

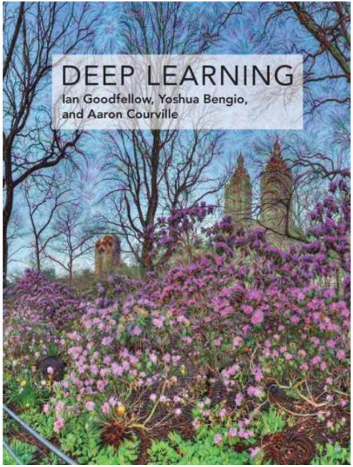
- Accurate prices
- Coherent groupings
- KPI lifts
- Correctly sorted images

Machine Learning



[techleer.com/articles/203-machine-learning-algorithm-backbone-of-emerging-technologies]

(Deep) Neural Networks



DEEP LEARNING

Ian Goodfellow, Yoshua Bengio, and Aaron Courville

[http://www.deeplearningbook.org/]

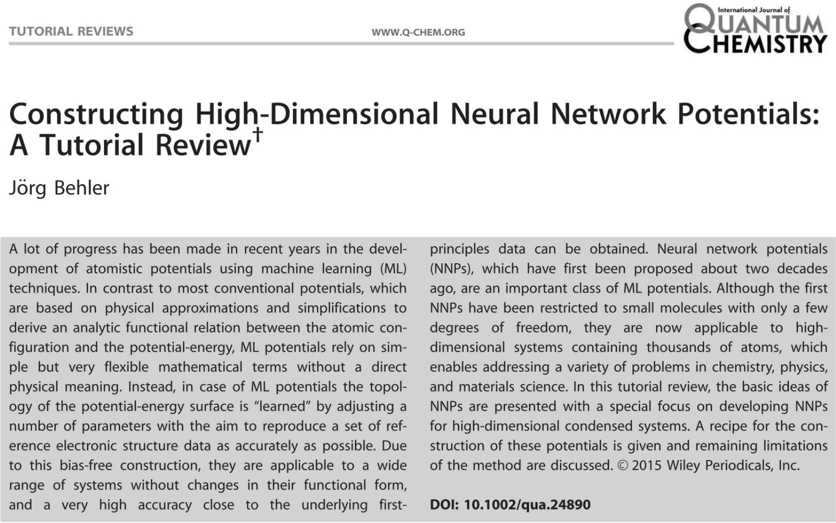
(Deep) Neural Networks

- High degree of flexibility and great expressive power
- Highly adaptive functional constructions in high-dimensions
- Useful as general surrogates for complex forward models
- As flexible parametrizations for phenomenological models
- Huge success in many fields
  - Speech or image recognition
  - Transportation autonomy
  - Myriad uses in progress, e.g. oil exploration, climate modeling
- NN flexibility and large data are crucial to established successes

ML Challenges/Opportunities in DOE Applications

- Physical considerations need to inform ML methods
  - Conservation laws, invariants, physical constraints
  - There is already some work in, e.g., chemistry, turbulence modeling
- Account for uncertainties: point predictions not good enough
- Overfitting and extrapolative errors much more dangerous
  - Need safeguards, such as Occam's razor, (Bayesian) model selection
- Approximation theory largely missing
- Quantification of predictive skill of ML systems under interpolation and extrapolation

Focus Paper



TUTORIAL REVIEWS WWW.Q-CHEM.ORG International Journal of QUANTUM CHEMISTRY

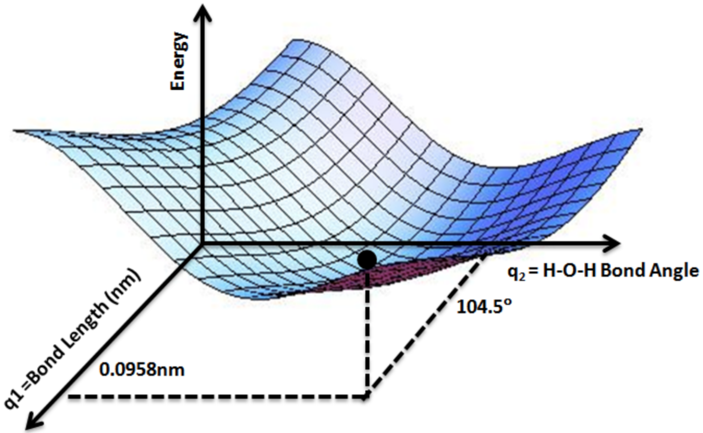
Constructing High-Dimensional Neural Network Potentials: A Tutorial Review[†]

Jörg Behler

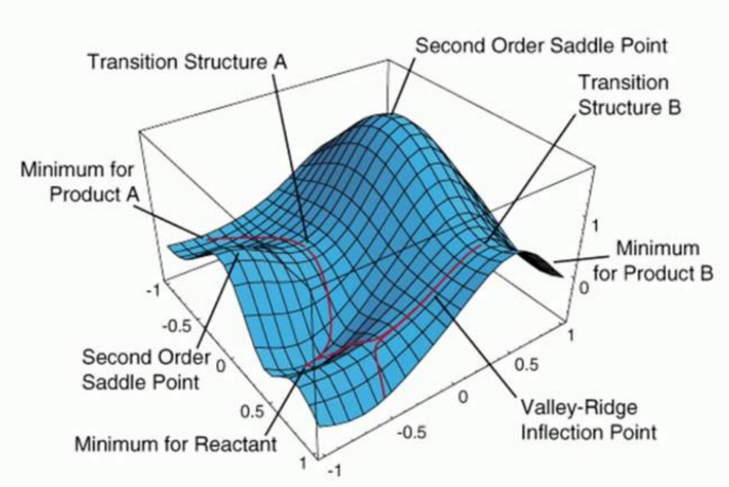
A lot of progress has been made in recent years in the development of atomistic potentials using machine learning (ML) techniques. In contrast to most conventional potentials, which are based on physical approximations and simplifications to derive an analytic functional relation between the atomic configuration and the potential-energy, ML potentials rely on simple but very flexible mathematical terms without a direct physical meaning. Instead, in case of ML potentials the topology of the potential-energy surface is “learned” by adjusting a number of parameters with the aim to reproduce a set of reference electronic structure data as accurately as possible. Due to this bias-free construction, they are applicable to a wide range of systems without changes in their functional form, and a very high accuracy close to the underlying first-principles data can be obtained. Neural network potentials (NNPs), which have first been proposed about two decades ago, are an important class of ML potentials. Although the first NNPs have been restricted to small molecules with only a few degrees of freedom, they are now applicable to high-dimensional systems containing thousands of atoms, which enables addressing a variety of problems in chemistry, physics, and materials science. In this tutorial review, the basic ideas of NNPs are presented with a special focus on developing NNPs for high-dimensional condensed systems. A recipe for the construction of these potentials is given and remaining limitations of the method are discussed. © 2015 Wiley Periodicals, Inc.

DOI: 10.1002/qua.24890

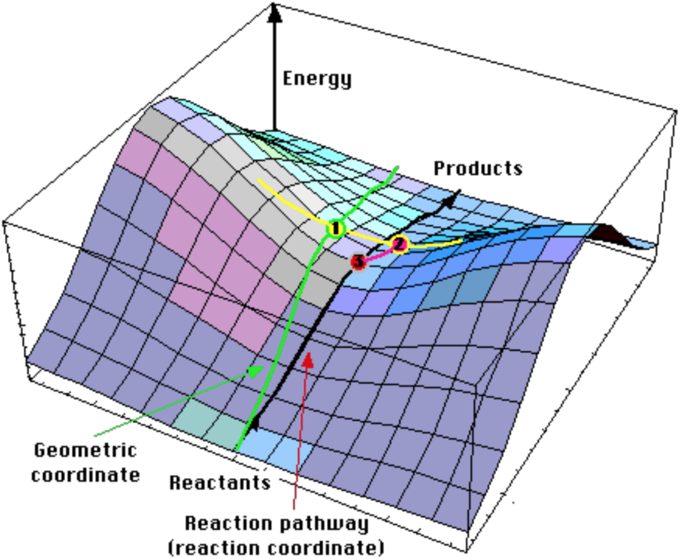
Potential Energy Surface



Potential Energy Surface



Potential Energy Surface

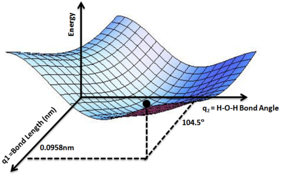


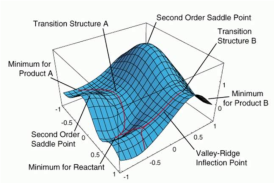
Main target: PES approximation

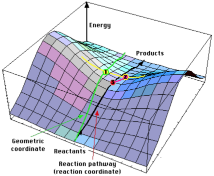
$$E = f(x)$$

x is coordinates/descriptors E is energy

- Accurate and fast surrogates for PES to replace QM computations for studies requiring many PES inquiries
  - saddle point search, transition paths, barrier heights
  - rapid assessment of reaction characteristics
  - automate the discovery of reactive pathways







PES

- Born-Oppenheimer apprx.: energy separation into electronic and nuclear components
  - Ground state PE is completely defined by the atomic positions, the nuclear charges, and the total charge of the system
- Exact solution of Schrödinger eqn impossible for all but smallest systems
- Electronic structure calculations: balance between efficiency and accuracy
  - Quantum mechanical computations
  - *ab initio*, density functional theory (DFT)
- There is clearly well-defined, but expensive relationship between atomic positions and potential energy
- Simplified forms are needed

Physical vs Mathematical PES Apprx's

- Simplified energy expressions based on physical considerations
- Central idea: decompose into low-dim bonding of 2-,3-,4-body
- Classical force fields, reactive potentials
- (Embedded) atom method
- Brenner potential, Tersoff potential, Stillinger-Weber potential, Lennard-Jones potential
- Accuracy limited by the imposed functional form

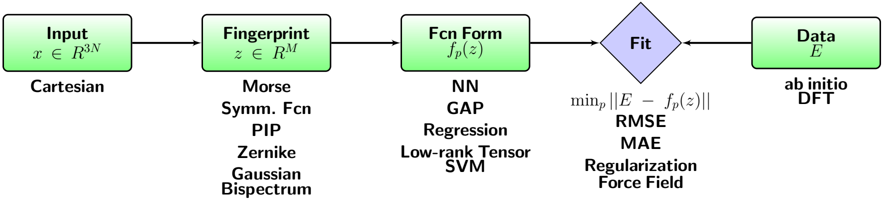
Physical vs **Mathematical** PES Apprx's

- also called Machine Learning (ML); No direct physical meaning
- Huge growth last decade
  - Weighted interpolation [Ischtwan 1994; Dowes, 2007-09; Maisuradze, 2009]
  - Permutationally invariant polynomials [Xie, 2010]
  - Gaussian processes [Bartok, Csanyi 2010-15; Mills, 2012; Rupp, 2013; Cui, 2016; Uteva, 2017; Guan, 2018; Schmitz, 2018]
  - Low-rank tensor expansions [Jackle, 1996; Baranov, 2015; Rai, 2017, 2018]
  - Support vector machines, kernel regression [Le, 2009; Balabin, 2011; Dral, 2017]
  - Neural networks (NN) [Blank, 1995; Tai No, 1997; Prudente, 1998; Lorenz, 2004; Witkoskie, 2005; Manzhos, 2006-09; Malshe, 2008; Le, 2009] [Behler, 2010-16; Handley, 2010, 2014; Jiang, 2013; Li, 2013; Dolgirev, 2016; Khorshidi, 2016; Peterson, 2016; Carr, 2016; Kolb, 2016; Shao, 2016; Chmiela, 2017; Cubuk, 2017; McGibbon, 2017; Smith, 2017; Schutt, 2017; Yao, 2017; Hajinazar, 2017; Bereau, 2018; Lubbers, 2018; Unke, 2018; Wang, 2018; Natarajan, 2018; Zhang, 2018; Onat, 2018]

ML Potentials

- Particularly useful when:
  - Many interaction types: covalent, ionic, metallic bonding, dispersion
  - Uncommon environments: amorphous systems, phase transitions
  - Making/breaking bonds
- Requirements:
  - Accurate
  - Fast-to-evaluate
  - Analytic derivatives
  - Systematic improvement
  - Reactive, i.e. enable making/breaking bonds
  - High-dimensional
  - Transferable/generalizable to unseen atomic configurations
  - Translational, rotational, permutational invariance

Main ingredients for supervised ML



Input $x \in R^{3N}$

Cartesian

Fingerprint $z \in R^{M}$

Morse
Symm. Fcn
PIP
Zernike
Gaussian
Bispectrum

Fcn Form $f_p(z)$

NN
GAP
Regression
Low-rank Tensor
SVM

Fit

$\min_p ||E - f_p(z)||$

RMSE
MAE
Regularization
Force Field

Data E

ab initio
DFT

- Training data $(\boldsymbol{x}_i, E_i)$ for $i = 1, \ldots, S$, and $\boldsymbol{x}_i \in R^{3N}$
  - *ab initio*, DFT
- Input representation, aka fingerprint, aka descriptor

$$(\boldsymbol{z}_i, E_i)$$

- Parameterized functional form of the approximation class

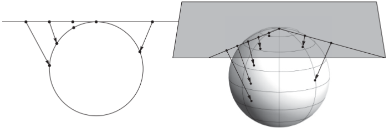
$$f_p(\boldsymbol{z})$$

- Loss function

$$\min_p \sum_{i=1}^{S} \left(E_i - f_p(\boldsymbol{z}_i(\boldsymbol{x}))\right)^2$$

Representation

- Internal coordinates,
  $\boldsymbol{x}_i \rightarrow z_i \equiv$(bond length, bond angle, and dihedral angle).
  Dimensionality $3N \rightarrow 3N-6$
- Morse variables, [Qu,Yu, Bowman, ARPC, 2018]:
  $N(N-1)/2$ variables, from internuclear distances $z_{jk} = e^{-||\boldsymbol{x}_j - \boldsymbol{x}_k||/\gamma}$
- Invariant low-order polynomials [Xie, 2010]. Think of $x_i^2 y_i \rightarrow x_i^2 y_i + x_i y_i^2$
- Bispectrum [Bartok, 2010]

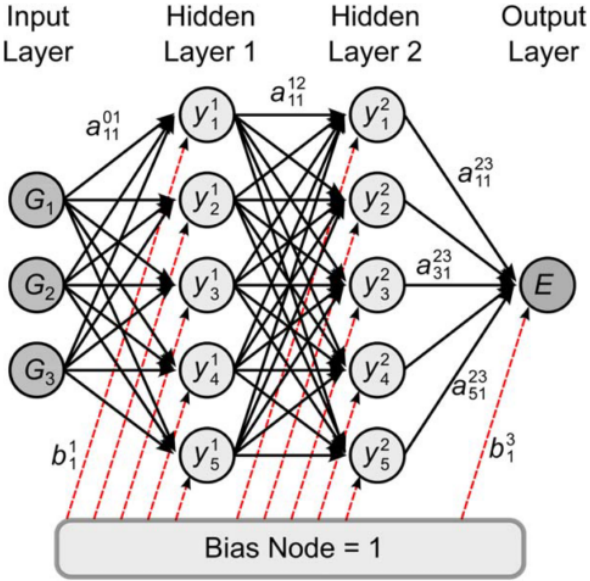


- **Symmetry functions** [Behler and Parrinello, 2007; Behler, 2011]
  - Details coming

Functional Forms

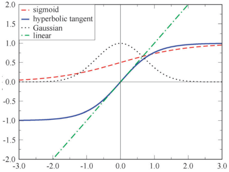
- Sparse Regression $f(x) = \sum_{k=0}^{K} c_k \Psi_k(x)$
  - Polynomial basis
  - Radial basis functions
- Low-Rank Tensor Expansion $f(x) = \sum_{r=1}^{R} c_r \prod_{j=1}^{d} \Psi_{rj}(x_j)$
  - Canonical format
  - Tensor-train format
- Gaussian processes $P(f|\mathcal{D})$
  - Hierarchical correction
  - Flexible kernels
- Neural Networks $f(x) = ...W_3\sigma(W_2\sigma(W_1x + b_1) + b_2) + b_3$
  - **Multilayer Feed-Forward NN**
  - Convolutional NN
  - Recurrent NN
  - ...

(Feed Forward) Neural Network



NN is a highly parameterized functional form

- Linear function $y = \boldsymbol{w}\boldsymbol{x} + b$
- Many of those $\boldsymbol{y} = \boldsymbol{W}_1\boldsymbol{x} + \boldsymbol{b}_1$
- Apply nonlinear activation function $\boldsymbol{\sigma}(\boldsymbol{W}_1\boldsymbol{x} + \boldsymbol{b}_1)$
- Repeat $\boldsymbol{\sigma}(\boldsymbol{W}_2\boldsymbol{\sigma}(\boldsymbol{W}_1\boldsymbol{x} + \boldsymbol{b}_1) + \boldsymbol{b}_2)$

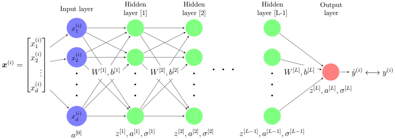


- Another layer: $\boldsymbol{\sigma}(\boldsymbol{W}_3\boldsymbol{\sigma}(\boldsymbol{W}_2\boldsymbol{\sigma}(\boldsymbol{W}_1\boldsymbol{x} + \boldsymbol{b}_1) + \boldsymbol{b}_2) + \boldsymbol{b}_3)$
- I told you so: a parameterized function

$$E = f_p(\boldsymbol{x})$$

with $p = (\boldsymbol{W}_1, \ldots, \boldsymbol{W}_L, \boldsymbol{b}_1, \ldots, \boldsymbol{b}_L)$

- Number of parameters: $L(N+1)N$

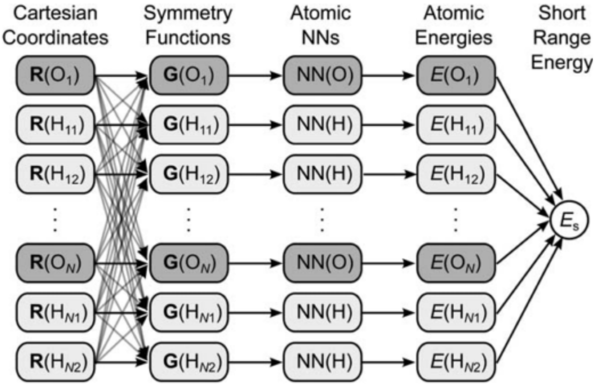

Input layer
Hidden layer [1]
Hidden layer [2]
Hidden layer [L-1]
Output layer
$\boldsymbol{x}^{(i)} = \begin{bmatrix} x_1^{(i)} \\ x_2^{(i)} \\ \vdots \\ x_d^{(i)} \end{bmatrix}$
$x_1^{(i)}$
$x_2^{(i)}$
$x_d^{(i)}$
$W^{[1]}, b^{[1]}$
$W^{[2]}, b^{[2]}$
$W^{[L]}, b^{[L]}$
$\hat{y}^{(i)} \longleftrightarrow y^{(i)}$
$z^{[L]}, a^{[L]}, \sigma^{[L]}$
$a^{[0]}$
$z^{[1]}, a^{[1]}, \sigma^{[1]}$
$z^{[2]}, a^{[2]}, \sigma^{[2]}$
$z^{[L-1]}, a^{[L-1]}, \sigma^{[L-1]}$

Neural network potentials

- Pros
  - High accuracy
  - Much cheaper than electronic structure calculations
  - Black-box: no-knowledge of the functional form of PES
  - No system-specific modifications required
  - Forces often computable by chain rule
- Cons
  - Slower than parameterized, classical force fields
  - Extrapolation questionable
  - Overfitting happens
  - Default implementation does not allow incremental addition of atoms
  - Training difficult
  - Low-to-moderate dimensions

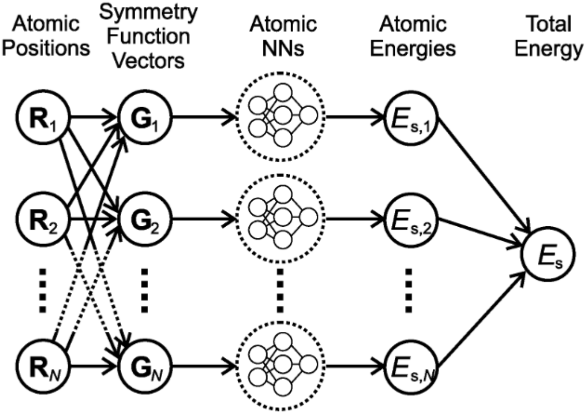
High-dimensional NN for PES [Behler-Perrinello, 2011]

- Map Cartesian coordinates to symmetry descriptors
- Write the total energy as a sum over all atoms $E_S = \sum_{i=1}^{N} E_i$
- Separate NN architecture for each atom



High-dimensional NN for PES [Behler-Perrinello, 2011]

- Map Cartesian coordinates to symmetry descriptors
- Write the total energy as a sum over all atoms $E_S = \sum_{i=1}^{N} E_i$
- Separate NN architecture for each atom



Symmetry Functions

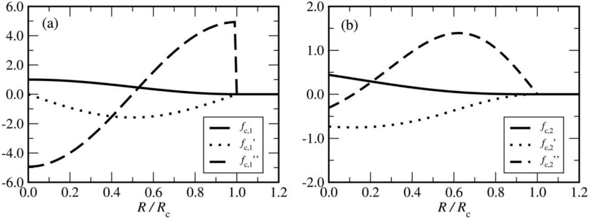
- Transformed coordinates, which are many-body functions of all atomic positions inside cutoff spheres
- Called symmetry functions for historic reasons
- Structurally equivalent representations of the system must give rise to the same set of coordinates
- Basic requirements:
    - rotational and translational invariance
    - invariance with respect to the permutation of atoms of the same element
    - provide a unique description of the atomic positions
    - constant number of function values independent of the number of atoms in the cutoff spheres

Symmetry Functions

- Cutoff functions

$$f_{c,1}(R_{ij}) = \begin{cases} 0.5\left(\cos\left(\frac{\pi R_{ij}}{R_c}\right)+1\right) & \text{for } R_{ij} \leq R_c \\ 0.0 & \text{otherwise} \end{cases}$$

$$f_{c,2}(R_{ij}) = \begin{cases} \tanh^3\left(1-\frac{R_{ij}}{R_c}\right) & \text{for } R_{ij} \leq R_c \\ 0.0 & \text{otherwise} \end{cases}$$



- Sym. fcns depend on the positions of *all* atoms inside the cutoff spheres. Therefore, in contrast to internal coordinates like bond lengths, their numerical values are not always easy to illustrate.

Symmetry Functions - Practical Recipes

- Build up for each atom
  - $G_i^1 = \sum_{j=1}^{N_{atom}} f_c(R_{ij})$
  - $G_i^2 = \sum_{j=1}^{N_{atom}} e^{-\eta(R_{ij}-R_s)^2} f_c(R_{ij})$
- Add angular dependence
  - $G_i^3 = 2^{1-\zeta} \sum_{k\neq i,j} \left[(1+\lambda\cos\theta_{ijk})^\zeta e^{-\eta(R_{ij}^2+R_{ik}^2+R_{jk}^2)} f_c(R_{ij}) f_c(R_{ik}) f_c(R_{jk})\right]$
  - $G_i^4 =$ same but without R_{jk}
- Many ‘free’ parameters : R_c, η, R_s, ζ, λ
- Transferability is a challenge
  - Weights are optimized for a fixed set of symmetry functions, and for a specific system
  - Consider symmetry functions as part of the NN
- Adaptive construction of symmetry functions
  - Quite empirical: check the range of values, de-correlate, check the forces, parsimony
  - Huge room for improvement

Preconditioning

- Scale the inputs to NN, i.e. symmetry fcn outputs
- Choice of initial NN weights
- Output scaling

Symmetry Functions and Forces

- Force is the negative derivative w.r.t Cartesian coordinate

$$F_x = -\frac{\partial E}{\partial x} = -\sum_{j=1}^{N_{atom}} \frac{\partial E_j}{\partial x} = -\sum_{j=1}^{N_{atom}} \sum_{\mu=1}^{N_{s,j}} \frac{\partial E_j}{\partial G_{j\mu}} \frac{\partial G_{j\mu}}{\partial x}$$

- Force acting on an atom depends on the positions of the atoms being as far as $2R_c$ away

- Forces also useful for training and validation

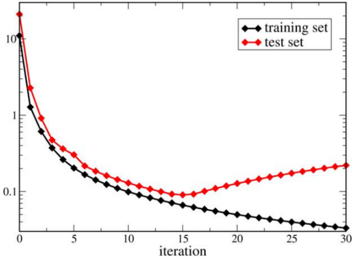
Loss Function and Training NNs: challenges

Training is an optimization problem to minimize the loss function

$$\min_p J(p) = \min_p \sum_{i=1}^{S} \left(E_i - f_p(\boldsymbol{x}_i)\right)^2$$

... in this case, x_i are the outputs of symmetry functions, and $p = (\boldsymbol{W}, \boldsymbol{b})$ are the NN parameters to tune. Training samples: $(\boldsymbol{x}_i, E_i)$

- Actually, you want to minimize $||E(x) - f_p(x)||$, but have to work with a finite sample estimate of it
- Training done via stochastic gradient descent and backpropagation
- Training error vs testing error: extrapolation challenge
- Generalization error, well-known challenge in NN: use regularization



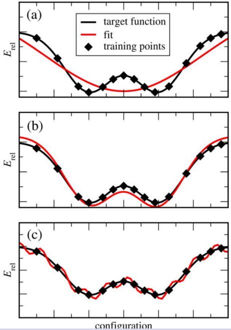
Loss Function and Training NNs: challenges

Training is an optimization problem to minimize the loss function

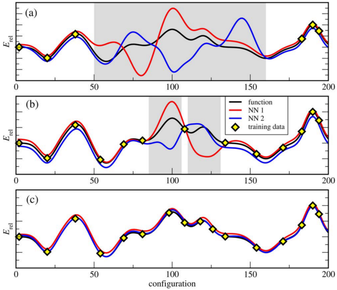
$$\min_p J(p) = \min_p \sum_{i=1}^{S} (E_i - f_p(\boldsymbol{x}_i))^2$$

... in this case, x_i are the outputs of symmetry functions, and $p = (\boldsymbol{W}, \boldsymbol{b})$ are the NN parameters to tune. Training samples: $(\boldsymbol{x}_i, E_i)$

- Actually, you want to minimize $||E(x) - f_p(x)||$, but have to work with a finite sample estimate of it
- Training done via stochastic gradient descent and backpropagation
- Training error vs testing error: extrapolation challenge
- Generalization error, well-known challenge in NN: use regularization



Training challenges and regularization



- Check inputs to ensure training inputs (i.e. symmetry fcn outputs) are within a range of validity
- Incremental increase of training set
- Regularization
  - Parameter norm penalties
  - Inclusion of forces in the loss function
  - Early stopping
  - Multiple-fit validation
  - Ensemble methods
  - Dropout

Software

- **ænet** The Atomic Energy Network; Artrith et. al
  `http://ann.atomistic.net/`; Fortran/C/(some) Python;
  Neural Network
- **MLatom** A Package for Atomistic Simulations with Machine Learning; Dral et. al
  `http://mlatom.com/`; Pre-compiled binaries on Linux;
  Kernel Ridge Regression
- **Amp** Atomistic Machine-learning Package; Andrew Peterson and Alireza Khorshidi
  `https://amp.readthedocs.io/`; Python, some Fortran, ASE;
  Neural Network
- **MLIP** Machine Learning of Interatomic Potentials; Shapeev et. al
  `http://gitlab.skoltech.ru/shapeev/mlip/`; C++, LAMMPS integration;
  Moment tensor potential (MTP), active learning
- **MSA** Monomial Symmetrization for PES Fitting; Bowman et. al
  `https://scholarblogs.emory.edu/bowman/msa/`; Fortran/C++/Python;
  Permutationally invariant polynomials
- **DeePMD-kit** Deep learning package for many-body PE and MD; Wang and E
  `https://github.com/deepmodeling/deepmd-kit`; Python/C++/Tensorflow;
  Neural Network
- **GAP** Gaussian Approximation Potentials; Csanyi and Bartok
  `http://www.libatoms.org/`; Fortran;
  Gaussian processes

Summary

- Mathematical PES clearly gaining momentum and work well
    - Any PES-intensive study, e.g. reaction pathway search
- Major decisions and opportunities
    - Training data selection
        - Need to include a large range of structures
        - Adaptive addition possible
    - Input representation
        - Selection of invariance-constraining representations
        - Automated model selection for optimal symmetry function choice
    - NN architecture selection
        - Plain feed-forward NN so far
    - Loss function selection
        - Inclusion of forces
        - Regularization penalties; soft and hard constraints
        - Goal-oriented
    - Training/validation procedure

Literature Thank you!

- J. Behler, “Constructing High-Dimensional Neural Network Potentials: A Tutorial Review”, *International Journal of Quantum Chemistry*, 115(16), 2015.
- J. Behler, “Perspective: Machine learning potentials for atomistic simulations”, *Journal of Chemical Physics*, 145(17), 2016.
- J Behler, “Representing potential energy surfaces by high- dimensional neural network potentials” *Journal of Physics: Condensed Matter*, 26(18), 2014.
- J. Behler, “Neural network potential-energy surfaces in chemistry: a tool for large-scale simulations”, *Physical Chemistry Chemical Physics*, 13(40), 2011.
- J. Behler, “Atom-centered symmetry functions for constructing high-dimensional neural network potentials”, *Journal of Chemical Physics*, 134(7), 2011.
- J. Behler and M. Parrinello, “Generalized Neural-Network Representation of High-Dimensional Potential-Energy Surfaces”, *Physical Review Letters*, 98(14), 2007.

- Z. Xie, J. Bowman, “Permutationally Invariant Polynomial Basis for Molecular Energy Surface Fitting via Monomial Symmetrization”, *Journal of Chemical Theory and Computation*, 6, 2010.
- A. Bartok, G. Csànyi, “Gaussian Approximation Potentials: A Brief Tutorial Introduction”, *International Journal of Quantum Chemistry*, 115, 2015.
- A. Khorshidi, A. Peterson, “Amp: A modular approach to machine learning in atomistic simulations”, *Computer Physics Communications*, 207, 2016.

- P. Rai, K. Sargsyan, H. Najm, M. Hermes, S. Hirata, “Low-rank canonical-tensor decomposition of potential energy surfaces: Application to grid-based diagrammatic vibrational Green’s function theory”, *Molecular Physics*, 115(17-18), 2017.
- P. Rai, K. Sargsyan, H. Najm, “Compressed sparse tensor based quadrature for vibrational quantum mechanics integrals”, *Computer Methods in Applied Mechanics and Engineering*, 336, 2018.