How Machine Learning and Chemistry Meet in the Quantum Chemistry Field.

Johnatan Mucelini

Advisor: Juarez L. F. Da Silva

August, 2020

Quantum Chemistry

Obtaining Properties with Quantum Chemistry:

System: $\{\mathbf{R}_i, Z_i\}$, N
$\mathbf{R}_i$: nuclear positions
Z_i: nuclear charge
N: # electrons

Mapping into energy: $\{\mathbf{R}_i, Z_i\} \xrightarrow{\Psi} E$
QC equations:
WF: Schrödinger equation $\hat{\mathbf{H}}\Psi = E\Psi$
DFT: Kohn-Sham equation $\hat{\mathbf{H}}_{eff}\phi^i_{ks} = e^i\phi^i_{ks}$

$$\rho = \sum_i \left|\phi^i_{ks}\right|^2$$

Mapping into other properties: $E, \rho \rightarrow property$

$$F = \frac{\partial E}{\partial R_i} \qquad Q_i = \int_{basis\ i} \rho\, d\tau - Z_i$$

Problems related to solving QC equation:

- Exponential scaling of computational cost for large systems:
  DFT (cheap method): approximately $O(N_e^3)$
  DFT (VASP): close to $O(N^2_{atoms} \ln N_{atoms})$
  Computational Materials Science 6 (1996) 15-50
- In global optimizations and high-throughput screening:
  - QC equations are extensively solved.
  - Big data complicate some tasks, such as extract representative sets.

Dream: functions that map systems ($\{\mathbf{R}_i, Z_i\}$) into the energy and properties directly: (without QC equations)

$$f(\{\mathbf{R}_i, Z_i\}) = E$$

Representation, how ML "see" QC

ML "see" QC system (take the input) as a vector of values (features) that represent the system.

$$\mathbf{x} = [x_1, x_2, ...]$$

Intuitive descriptors:

- $\{\mathbf{R}_i, Z_i\}$.
- # bonds, avarege/ranked bond distance.

Qualities of a good representation:

- Invariance to the rotation, translation, and homo-nuclear permutational.
- Non-degenerate.
- Unique.
- Size extensively.

Smooth Overlap of Atomic Positions (SOAP) Neighbor density function:

$$\rho^i = \sum_j \exp\left(-\sigma|\mathbf{r} - \mathbf{r}_j|^2\right) = \sum_{nlm} c^i_{nlm} g_n(r) Y_{lm}(\hat{\mathbf{r}})$$

where $g_n(r)$ is a orthonormal radial basis function.

ρ^{X} X═Y═Y Y═X═Y

From ρ^{X} one can derivate the rotational invariants

$$x_{nn'l} = \sum_m c_{nlm} (c_{n'lm})^*$$

which can be employed as elements of a descriptor vector $\mathbf{x}^{\mathrm{X}}$, and $\mathbf{x} = [\mathbf{x}^{\mathrm{X}}, \mathbf{x}^{\mathrm{Y}}, \mathbf{x}^{\mathrm{Y}'}]$.

Paradigms, how ML learn from QC examples/data

Two main classes: **Supervised** and **Unsupervised learning**

Supervised learning, simple regression:
Input: $\mathbf{X}$ (features), $\mathbf{y}$ (target/label)

$$\mathbf{X} = \begin{bmatrix} \mathbf{x^1} \\ \mathbf{x^2} \\ \dots \\ \mathbf{x^N} \end{bmatrix} = \begin{bmatrix} x_1^1 & x_2^1 & \dots & x_M^1 \\ x_1^2 & x_2^2 & \dots & x_M^2 \\ \dots & \dots & \dots & \dots \\ x_1^N & x_2^N & \dots & x_M^N \end{bmatrix} \quad \mathbf{y} = \begin{bmatrix} y^1 \\ y^2 \\ \dots \\ y^N \end{bmatrix}$$

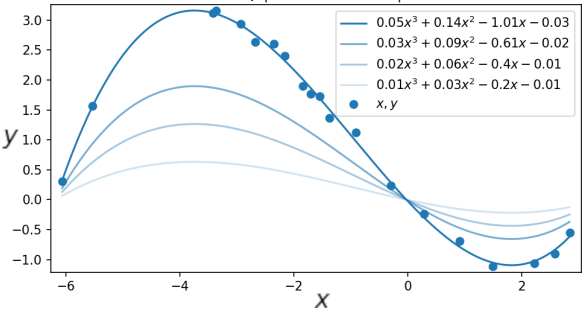
$h_\theta(\mathbf{x}^i)$: model (map $\mathbf{x}^i \rightarrow y^i$)

$J(\mathbf{y}, h_\theta(\mathbf{X}))$: cost function

$$J = \sum_i^N \left| y^i - h_\theta(\mathbf{x}^i) \right|^2$$

Learning task: $\text{argmin}_\theta \{ l_n(\mathbf{y}, h_\theta(\mathbf{X})) \}$

Input: $\{y^i, x^i\}$ ($\mathbf{x}^i$ present 1 element/feature)
Polinomial model: $h_\theta(x) = \theta_0 + \theta_1 x^1 + \theta_2 x^2 + \theta_3 x^3$
Cost function: $J = \sum_i \left| y^i - h_\theta(x^i) \right|^2$



Training set: $X^{train} = \mathbf{X}[70\%]$
Test set: $X^{test} = \mathbf{X}[30\%]$

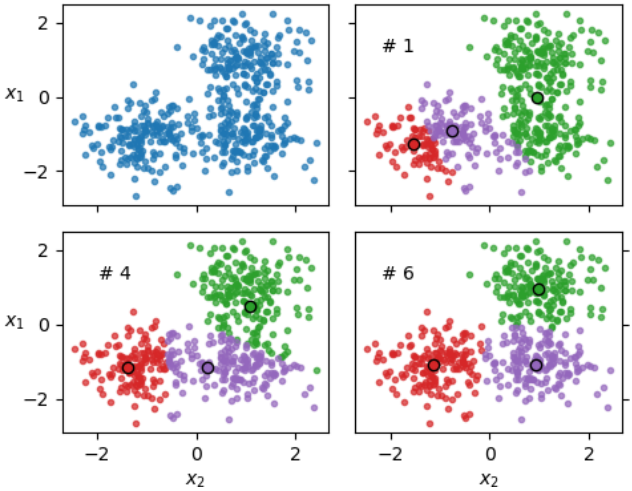
Paradigms, how ML learn from QC examples/data

Two main classes: **Supervised** and **Unsupervised learning**

Unsupervised learning, KMeans clusterization:
Input: $\mathbf{X}$ (features)

$$\mathbf{X} = \begin{bmatrix} \mathbf{x^1} \\ \mathbf{x^2} \\ \dots \\ \mathbf{x^N} \end{bmatrix} = \begin{bmatrix} x_1^1 & x_2^1 & \dots & x_M^1 \\ x_1^2 & x_2^2 & \dots & x_M^2 \\ \dots & \dots & \dots & \dots \\ x_1^N & x_2^N & \dots & x_M^N \end{bmatrix}$$

For a given number of $\mathbf{C}_i \in \Re^M$
Algorithm:
0) Initialize $\mathbf{C}_i$.
1) $\mathbf{x}^i$ is associated with the closest $\mathbf{C}_j$.
2) $\mathbf{C}_j$ is the mean point of the $\mathbf{x}^i$ associated.
3) Repeat steps 1 and 2 until convergence.
Problems: Fall in local minimums, equal clusters sizes, sphere shape

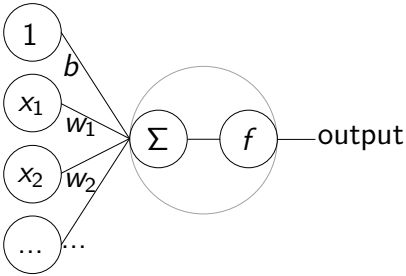


Machine Learning: Neural Networks

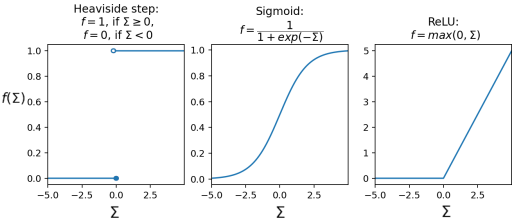
The Perceptron:

$$f(\mathbf{x}) = \begin{cases} 1 & \text{if } \mathbf{w} \cdot \mathbf{x} + b \geq 0 \\ 0 & \text{else} \end{cases}$$

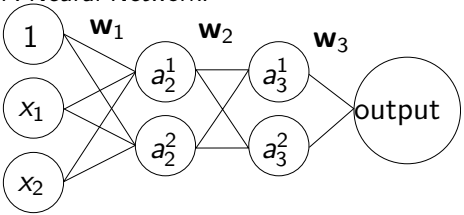
The Neural Network node:

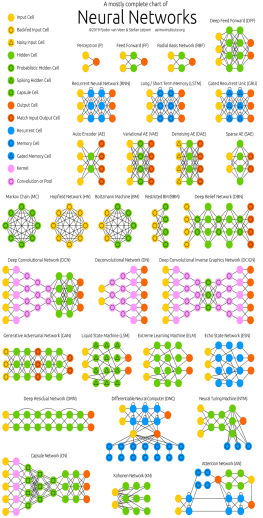


Activation functions:



A Neural Network:





Gaussian Process Regression

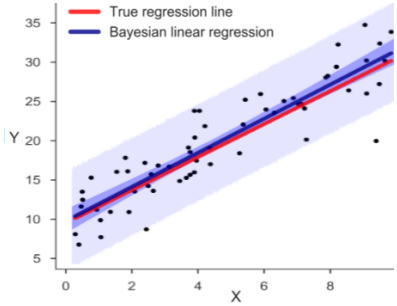
Bayasian Linear Regression

$$h(\mathbf{x}) = \mathbf{x}^T\mathbf{w}, \quad y = h(\mathbf{x}) + \varepsilon, \quad \varepsilon \sim N(0, \sigma_n^2)$$

$$p(\mathbf{w}|\mathbf{y}, \mathbf{X}) = \frac{p(\mathbf{y}|\mathbf{X}, \mathbf{w})p(\mathbf{w})}{p(\mathbf{y}|\mathbf{X})}, \quad p(\mathbf{w}) \sim N(0, \Sigma)$$

where $A = \sigma^{-2}\mathbf{X}\mathbf{X}^T + \Sigma^{-1}$. Predicting h for $\mathbf{x}'$:

$$p(h(\mathbf{x}')|\mathbf{x}', \mathbf{X}, \mathbf{y}) = N(\mathbf{x}'^T A^{-1}\mathbf{X}\mathbf{y}, \mathbf{x}'^T A^{-1}\mathbf{x}')$$



Seppe vanden Broucke et al., medium.com, 2018

The Kernel Trick:

$\phi(\mathbf{x})$ maps $\mathbf{x}$ into another space. Defining $\psi(\mathbf{x}) = \Sigma^{1/2}\phi(\mathbf{x})$ we obtain a dot product $\psi(\mathbf{x}^i) \cdot \psi(\mathbf{x}^j) = \phi(\mathbf{x}^i)^T\Sigma\phi(\mathbf{x}^j) = k(\mathbf{x}^i, \mathbf{x}^j) = k_{ij'} \in \Re$.

Gaussian Process Regression

Applying Φ on $\mathbf{x}$, $h(\mathbf{x}) = \phi(\mathbf{x})^T\mathbf{w}$, one found the result with k intead ϕ:

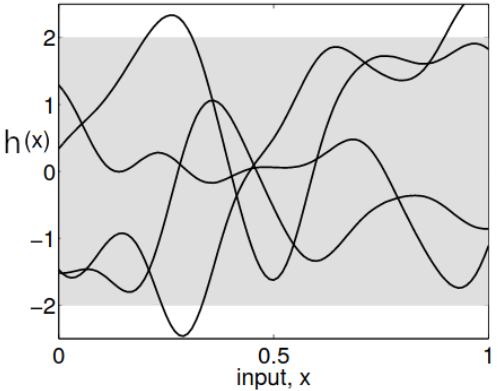
$$p(h|\mathbf{x}', \mathbf{X}, \mathbf{y}) = N(\phi'^T\Sigma\Phi(K + \sigma^2 I)^{-1}\mathbf{y}, S)$$

$$S = \phi'^T\Sigma\phi' - \phi'^T\Sigma\Phi(K + \Sigma^2 I)^{-1}\Phi^T\sigma\phi')$$

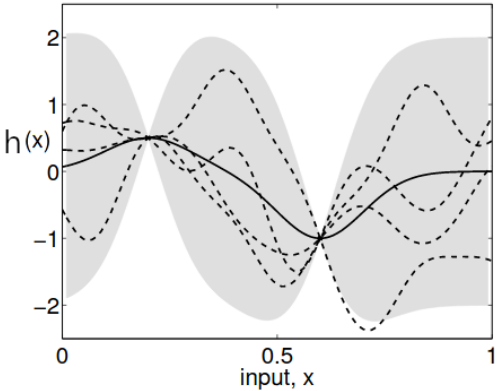
where $K = \Phi^T\Sigma\Phi$ and $\Phi = \Phi(\mathbf{X})$.

$$K = \begin{bmatrix} k_{0,0} & k_{0,1} & \dots & k_{0,N} \\ k_{1,0} & k_{1,1} & \dots & k_{1,N} \\ \dots & \dots & \dots & \dots \\ k_{N,0} & k_{0,1} & \dots & k_{0,N} \end{bmatrix}$$

Gaussian Process Regression



(a), prior



(b), posterior

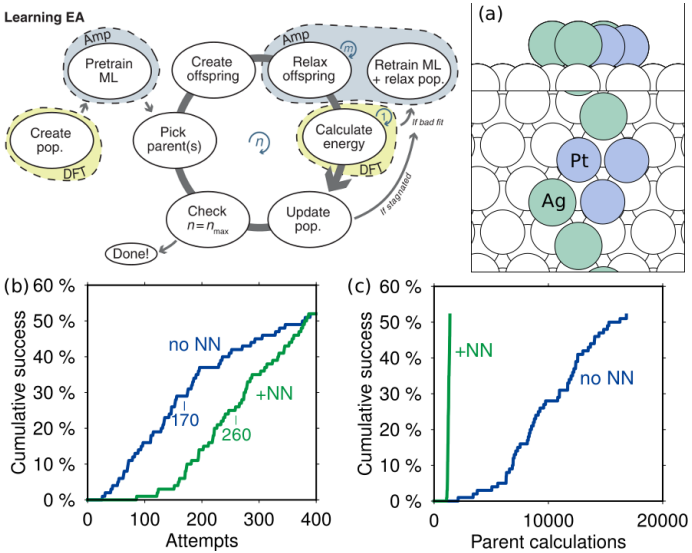
C. E. Rasmussen & C. K. I. Williams, Gaussian Processes for Machine Learning, 2006.

Examples: Kolsbjerg et al., 2018

ML + Evolutionary algorithm + QC:

- ML Model: NN (2 hidden layers, 5 nodes each).
- Representation: Gaussian (G2 and G4) descriptor.
- Finding the global minimum structure with 50% certainty per run:
  - no NN: ∼170 generations.
  - +NN: ∼260 generations.
  - +NN runs successful is delayed.
- Successful runs:
  - no NN: ∼8900 QC calculations.
  - +NN: ∼260 QC calculations.
  - +NN reduce # calculations by ×40.

Phys. Rev. B, 97, 2018, 195424



Examples: Podryabinkin et al., 2019

ML + Evolutionary algorithm + QC:

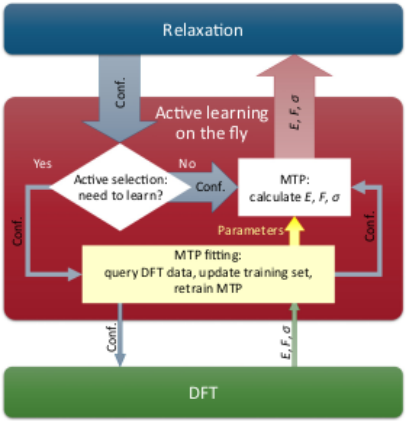
- ML Model: Moment Tensor Potentials (MTP).

$$E = \sum_i V_i$$

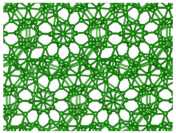
$$V_i = \sum_j w_j B_j(\mathbf{u}_i)$$

- Evolutionary algorithm: USPEX.
- Active learning: prediction "extrapolation".
- Time spend to find the 6 lowest-energy structures reduce 1-4 orders.
- Efficiency is improved with pre-training.

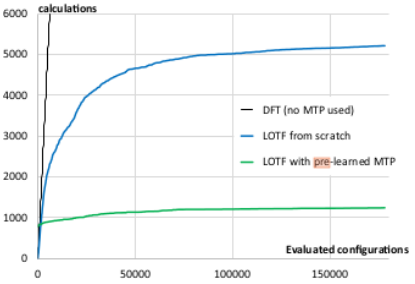
- Configuration relaxation inside the evolutionary algorithm:



- Material: Boron allotropes

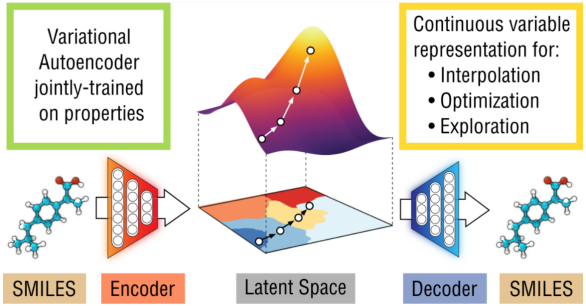


β-boron approximant
$E^{DFT} = -6.704$ eV/atom,
Atoms: 106,
Space group: $P1$,
Core-hours: $7 \cdot 10^3$ AL-MTP vs. $6.6 \cdot 10^7$ DFT
$|E^{DFT} - E^{MTP}| = 10.1$ meV/atom

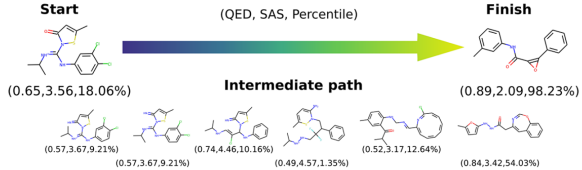


Phys. Rev. B 99, 064114 (2019)

Examples: Gómez-Bombarelli et al., 2018



- Variational Autoencoder (VAE): a NN used for dimensionality reduction and generative processes.
- SMILES: string representation of molecules.
- Predictor model: Gaussian Process
- Autoencoder training: 250k molecules (ZINC)
- Encodes approximately: 7.5M molecules



ACS Cent. Sci. 4, 2018, 268-276

BRIEF COMMUNICATION
https://doi.org/10.1038/s41587-019-0224-x

nature biotechnology

Deep learning enables rapid identification of potent DDR1 kinase inhibitors

Alex Zhavoronkov[1*], Yan A. Ivanenkov[1], Alex Aliper[1], Mark S. Veselov[1], Vladimir A. Aladinskiy[1], Anastasiya V. Aladinskaya[1], Victor A. Terentiev[1], Daniil A. Polykovskiy[1], Maksim D. Kuznetsov[1], Arip Asadulaev[1], Yury Volkov[1], Artem Zholus[1], Rim R. Shayakhmetov[1], Alexander Zhebrak[1], Lidiya I. Minaeva[1], Bogdan A. Zagribelnyy[1], Lennart H. Lee[2], Richard Soll[2], David Madge[2], Li Xing[2], Tao Guo[2] and Alán Aspuru-Guzik[3,4,5,6]

Nat. Biotechnol., 37, 2019, 1038-1040

Overview and Conclusions

- Quantum Chemistry.
  - High computational costs.
  - Many calculations are necessary.
- Machine Learning.
  - Representation.
    - How ML "see" chemistry structures.
  - Supervised:
    - Ex.: NN, gaussian process.
  - Unsupervised:
    - Ex.: clustering, VAE.
- Wide range of applications:
  - Ex.: Nanomaterials and Drug Design.
  - Commonly coupled with other algorithms.

Acknowledgments

CINE
QTNano
IQSC
CNPq
CAPES
FAPESP
UNIFESP
UFABC

Thanks for your Attention!

Supporting Slides