Data science for engineering & science

nanoHUB
1.6 M visitors,
16,000+ simulation users per year

Ale Strachan
strachan@purdue.edu
Purdue University
3 Technologies That Could Create Trillion-Dollar Markets Over the Next Decade

By Greg Satell  Updated April 21, 2019 / Original February 17, 2019

Sequential learning for materials discovery

Data science & machine learning

Quantum Mechanics

Molecular Dynamics

Components

Quantify the confidence in the prediction

Pareto front

nanoHUB

Cyber-infrastructure
Dimensionality reduction (Coarse graining & multiscale modeling)

Learning from data

Predictive models

Classification

Design of experiments

Goal achieved?

YES: SUCCESS

NO: Add data & restart

Query information source

Information acquisition function

Predictive model of state of knowledge

Existing data
nanoHUB: online simulations, data & more

- 620+ simulation tools
- 1,800+ contributors
- ~18,000+ annual simulation users
- 1,300,000+ simulations per year
- 1.6+ million visitors per year

www.nanohub.org/usage
Apps connected to powerful research codes

Designed for end users: instructors, students, domain experts

https://nanohub.org/tools/dftmatprop
Powered by Quantum Espresso

https://nanohub.org/tools/nanomatmech
Powered by LAMMPS

1. Use **Polymer Modeler** to create an amorphous polymer system

2. Visualize structure

3. MD simulations using **LAMMPS** on HPC resources

4. Post process results and plot

Publish your workflow with a few clicks. We containerize it for reproducibility.
Impact on research

2,300+ citations
5,600+ authors

2,000 use in research
1,600+ Outside NCN papers

- NCN-affiliated documents
- Non-NCN-affiliated documents
Impact on education

- Simulations in ALL top 50 US NWR Engineering Schools
- 2,897+ courses
  - 57,000+ students in classroom setting
  - 44% HBCUs, 37% of all HHE institutions, & 24% of all MSIs
- 170+ companies in nanoHUB courses
Publish for reproducibility and discoverability

- All tools have DOIs
- All tools are indexed by
  - Web of Science
  - Google Scholar

Self-serve publication process
Data Science & Machine Learning in Science & Engineering

Learning from data

Predictive models

Classification

Design of experiments

Cyber-infrastructure

Dimensionality reduction (Coarse graining & multiscale modeling)

Learning from data

Reagents

Products

Intermediates

Time (ps)

Goal achieved?

YES: SUCCESS

NO: Add data & restart

Query information source

Information acquisition function

Predictive model of state of knowledge

Existing data
Regression

Model:
- Simple functions, like polynomials
- Physics-based models (parameter calibration)
- Universal approximators, like artificial neural networks & random forests
  - Requires lots of data
Linear Regression – a materials example

Can we predict Young’s modulus based only on the melting temperature?

High bond strength – high melting temperature

High bond strength – high stiffness – high Young’s modulus

Melting temperature and Young’s modulus generally correlate to each other

Callister, 9th edition
• Melting temperature and stiffness are correlated

• Could we develop a more accurate model for Young’s moduli?
Neural Networks 101

\[
\begin{align*}
T_m & \\
\rho & \\
\text{Inputs} & \\
\text{Hidden layer} & \\
\text{Output} & \\
\text{MODEL} & \\
\end{align*}
\]

activation weights bias

\[
\begin{align*}
a_1 &= f_1(w_1 T_m + w_2 \rho + b_1) \\
a_2 &= f_1(w_3 T_m + w_4 \rho + b_1) \\
\hat{y} &= \text{Young's modulus} = f_3(w_5 a_1 + w_6 a_2 + b_3)
\end{align*}
\]

https://en.wikipedia.org/wiki/Biological_neuron_model
Neural Networks 101: activation functions & training

Objective

\[ \sum_{i=1}^{N_{\text{samples}}} (\hat{y} - y)^2 \]

Training to reproduce ground truth

\[ w_1 = w_1 - \alpha \frac{\partial (\text{Objective})}{\partial w_1} \]

Backpropagation
Prediction of Young’s moduli
How do we judge if the model has learnt all that it could?

- **Underfitting** – model hasn’t learnt all the trends in the training data
- **Overfitting** – model has “memorized” data, ignoring the underlying trend

How do we train models that generalize well?

- Use a low enough learning rate
- Optimize number of adjustable parameters
- Monitor error on validation set as a measure of model’s ability to generalize

[https://keras.io/callbacks/#earlystopping](https://keras.io/callbacks/#earlystopping)
Data science driven models for physics simulations

Length

Time

lifetime

second

microsec

nanosec.

picosec.

femtosec.

Molecular dynamics

Quantum Mechanics

Mesoscale

Components

Uncertainty Quantification

Quantitative Predictions

nanometer

micron

mm
Molecular dynamics

Follow the dynamics (motion) of all the atoms in your material

Numerically solve classical equations of motion (Newton’s):

Approximation

\[ \vec{F}_i = m_i \vec{A}_i \]

or

\[ \dot{\vec{R}}_i = \frac{\vec{P}_i}{M_i} \]

\[ \dot{\vec{P}}_i = \vec{F}_i \]

Forces on atoms come from the interaction with other atoms:

\[ \vec{F}_i = -\vec{\nabla}_{R_i} V(\{R_j\}) \]

Approximated (in almost all cases)

Total potential energy

- Electronic structure calculation
- An empirical force field
Hot spot formation via large scale MD


120nm x 240 nm
~9 M atoms
Shock to deflagration transition

20 nm pore

40 nm pore

Temperature (K)

Area Fraction above 1700K

Time (ps)
Atomic positions

Symmetry functions describe local atomic environment

Neural network

Atomic energy and forces

\[ w_{G_i}^{\text{rad}} = \sum_{j \neq i} Z_j e^{-\frac{r_{ij}}{\sigma}} \]

\[ w_{G_i}^{\text{ang}} = 2^{1-\xi} \sum_{j \neq i \neq k} e^{-\eta (r_{ij} - r_{ik})} e^{-\eta (r_{jk})} e^{-\eta (r_{jk})} f_c (r_{ij}) f_c (r_{ik}) f_c (r_{jk}) \]

**Neural Network Reactive Force Fields**

Atomic energy and forces

MODEL
Neural Network Reactive Force Fields

Starting DB using Classical force field (ReaxFF & DFT)
NN training
Converged NN Potential
Potential
Starting DB using Classical force field (ReaxFF & DFT)
NN training
Converged NN Potential
Potential
Dynamic simulations using NN: New validation set
Elementary reactions & monomers
DFT single points calculation of validation sets
Analyzing and extracting elementary reactions & chemical species
DFT for Transition states of elementary reactions
Testing chemical & physical properties
Automated process & large system

Pilsun Yoo, Michael Sakano, Saaketh Desai, Mahbubul Islam, Peilin Liao, and A. Strachn npj Comp Mat Sci (under review)
Learning bond dissociation curves

![Graphs showing bond dissociation energies for various compounds](image)
Decomposition of RDX

Isothermal decomposition simulations

- $T_0 = 2000 \text{ K}$
- $T_0 = 2500 \text{ K}$

- Gen1.9 NNRF
- ReaxFF-2014
- ReaxFF-2018
- ReaxFF-Ig

- $N_2$
- $H_2O$
- $CO_2$
- $NO_2$
- $NO$
- HNCO/HCNO
- $NH_3$
- $NH_4$

$Gen1.9 \text{ NNRF} = 25.66 \ (kcal/mol)$
$ReaxFF-2014 = 23.51$
$ReaxFF-2018 = 18.73$
$ReaxFF-Iw = 11.96$
$ReaxFF-Ig = 11.87$
Data Science & Machine Learning in Science & Engineering

Learning from data
- Predictive models
- Classification
- Design of experiments
  - Existing data
  - Predictive model of state of knowledge
  - Query information source
  - Information acquisition function
- Dimensionality reduction
  - (Coarse graining & multiscale modeling)

Cyber-infrastructure
- nanoHUB
- CITRINE
- KIM
- WolframAlpha
- The Materials Project

Reagents
Intermediates
Products

Goal achieved?
YES: SUCCESS
NO: Add data & restart
Can ML help reduce the number of experiments to achieve a desired goal?

**Data science for design of experiments**

Existing data

Goal achieved?
YES: SUCCESS
NO: Add data & restart

Predictive model of state of knowledge

Query information source

Information acquisition function

Maximizing Li+ conductivity in solid oxides

Machine Learning Algorithms for Ionic Conductivity of LLZO-type Garnets

Overview
In this notebook, I introduce different regression methods for predicting the ionic conductivity of LLZO-type garnets. The dataset consists of 199 samples, with 198 of them being used for training and one for testing. The goal is to predict the ionic conductivity using various machine learning algorithms.

Import libraries
```python
import pandas as pd
df = pd.read_csv('llzo_data.csv')
```

The dataset contains the following columns:
- Chemical formula
- Preparation
- Reference
- Crystallographic structure
- Ionic conductivity
- Ionic conductivity units

The notebook explores the following algorithms:

1. Linear Regression
2. Decision Tree
3. Random Forest
4. K-Nearest Neighbors
5. Support Vector Machine
6. Gradient Boosting
7. XGBoost

Analysis and results are presented for each algorithm, showcasing their performance in predicting the ionic conductivity of LLZO-type garnets.
Find the best conductor with the fewest experiments

- Start with 10 data points (black) out of 100
- Train a ML model and evaluate on possible experiments
- Use acquisition functions to decide what experiment to reveal next
- Add result and iterate

Decision trees and random forests

- $\Delta H_f < 300$
  - $K = 1 \times 10^{-4}$

- $\Delta H_f < 340$
  - $K = 6 \times 10^{-5}$

- $\Delta H_f < 650$
  - $K = 3 \times 10^{-5}$
  - $K = 1 \times 10^{-6}$

<table>
<thead>
<tr>
<th>heat_of_formation</th>
<th>lattice_constant</th>
<th>melting_point</th>
</tr>
</thead>
<tbody>
<tr>
<td>284.9</td>
<td>4.09</td>
<td>1235.10</td>
</tr>
<tr>
<td>330.9</td>
<td>4.05</td>
<td>933.50</td>
</tr>
<tr>
<td>368.2</td>
<td>4.08</td>
<td>1337.58</td>
</tr>
<tr>
<td>337.4</td>
<td>3.61</td>
<td>1356.60</td>
</tr>
<tr>
<td>669.0</td>
<td>3.84</td>
<td>2683.00</td>
</tr>
</tbody>
</table>

Heat of formation

Ionic conductivity

300 340 650
Random forests are an ensemble of decision trees.

Choose training data for each tree randomly. Choose features for each tree randomly.

Bagging: Choose training data for each tree randomly.

Bootstrapping: Choose features for each tree randomly.

### Heat of Formation Dataset

<table>
<thead>
<tr>
<th>Property</th>
<th>Value 1</th>
<th>Value 2</th>
<th>Value 3</th>
<th>Value 4</th>
<th>Value 5</th>
<th>Value 6</th>
<th>Value 7</th>
<th>Value 8</th>
</tr>
</thead>
<tbody>
<tr>
<td>Heat of Formation</td>
<td>284.9</td>
<td>330.9</td>
<td>368.2</td>
<td>337.4</td>
<td>669.0</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Lattice Constant</td>
<td>4.09</td>
<td>4.05</td>
<td>4.08</td>
<td>3.81</td>
<td>3.84</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Melting Point</td>
<td>1235.10</td>
<td>933.50</td>
<td>1337.58</td>
<td>1356.60</td>
<td>2683.00</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Specific Heat</td>
<td>0.237</td>
<td>0.900</td>
<td>0.129</td>
<td>0.385</td>
<td>0.133</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Atomic Mass</td>
<td>107.8882</td>
<td>26.9815</td>
<td>196.9666</td>
<td>63.5460</td>
<td>192.2170</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Atomic Radius</td>
<td>1.60</td>
<td>1.25</td>
<td>1.35</td>
<td>1.35</td>
<td>1.35</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Electrical Resistivity</td>
<td>1.63e-8</td>
<td>2.70e-8</td>
<td>2.20e-8</td>
<td>1.72e-8</td>
<td>4.70e-8</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
From your browser go to link: https://nanohub.org/tools/citrinetools

Citrine Tools for Materials Informatics

Click on Launch Tool to begin
Finding best conductor with the fewest experiments

- Run the notebook online
- Adapt it to your specific problem

https://nanohub.org/tools/citrinetools
Learn faster exploring community data

Use non-numerical inputs for clustering Sankey diagrams to explore clusters

Find patterns in the data correlation analysis

Machine learning models

Detect outliers
Exploring DFT calculations *DFTMatProp App*

Output: energy-volume curve

- **Bulk modulus**
- **Equilibrium Vol**

[https://nanohub.org/tools/dftmatprop](https://nanohub.org/tools/dftmatprop)
Visually explore the simulation database

Sankey diagrams to explore simulation DB
Which inputs correlate to the outputs?

**Explore correlations between inputs & outputs**

<table>
<thead>
<tr>
<th>OUTs</th>
<th>INs</th>
</tr>
</thead>
<tbody>
<tr>
<td>output.bulkMod</td>
<td>input.ke_cutoff_potential</td>
</tr>
<tr>
<td>output.eqVol</td>
<td>input.convergence</td>
</tr>
<tr>
<td>output.bulkMod</td>
<td>1.0</td>
</tr>
<tr>
<td>output.eqVol</td>
<td>0.11</td>
</tr>
<tr>
<td>input.ke_cutoff_potential</td>
<td>-0.12</td>
</tr>
<tr>
<td>input.convergence</td>
<td>0.0</td>
</tr>
<tr>
<td>input.ke_cutoff</td>
<td>-0.02</td>
</tr>
<tr>
<td>input.volSteps</td>
<td>0.07</td>
</tr>
<tr>
<td>input.degauass</td>
<td>-0.0</td>
</tr>
<tr>
<td>input_z_size</td>
<td>-0.25</td>
</tr>
<tr>
<td>input_y_size</td>
<td>-0.25</td>
</tr>
<tr>
<td>input_x_size</td>
<td>-0.25</td>
</tr>
<tr>
<td>input.volMin</td>
<td>-0.07</td>
</tr>
<tr>
<td>input.volMax</td>
<td>-0.7</td>
</tr>
<tr>
<td>input.k_num</td>
<td>0.0</td>
</tr>
</tbody>
</table>

Outputs (equilibrium volume and bulk modulus) correlate with KE cutoff

Outputs (equilibrium volume and bulk modulus) correlate with number of k-points
Learn from previous community runs

Silicon

Convergence of bulk modulus with k-points and KE cutoff
$B_0=87.7$ GPa

Convergence of bulk modulus with k-points and KE cutoff
$V_0=40.8$ Å$^3$

Determine parameters required for a converged calculation
Learn more

Hands-on Data Science and Machine Learning Training

By Alejandro Strachan¹, Saaketh Desai²
1. Materials Engineering, Purdue University, West Lafayette, IN 2. Purdue University, West Lafayette, IN

Video podcast
Slides/Notes podcast

https://nanohub.org/resources/33245

Saaketh Desai
Zachary McClure
Michael Sakano
Juan Carlos Verduzco

Introduction to Jupyter Notebooks, Data Organization and Plotting (2nd offering)
21 Apr 2020 Online Presentations Contributor(s): Juan Carlos Verduzco Gastelum, Alejandro Strachan
This tutorial gives an introductory demonstration of how to create and use Jupyter notebooks. It showcases the libraries Pandas to manipulate and organize data with functionalities similar to those of Excel on python, and Plotly, a library used to create interactive plots for enhanced...

Repositories and Data Management (1st offering)
21 Apr 2020 Online Presentations Contributors: Zachary D McClure, Alejandro Strachan
This tutorial introduces database infrastructure and APIs for performing different scales of querying. You will learn how to access different suites of information from three prominent databases, and some advanced examples of data visualization and processing will be discussed. The Querying Data...

Hands-on Supervised Learning: Part 1 - Linear Regression and Neural Networks
22 Apr 2020 Online Presentations Contributor(s): Saaketh Desai
This tutorial introduces supervised learning via Jupyter notebooks on nanoHUB.org. You will learn how to setup a basic linear regression in a Jupyter notebook and then create and train a neural network. The tool used in this demonstration is Machine Learning for Materials Science:

Hands-on Supervised Learning: Part 2 - Classification and Random Forests (1st offering)
24 Apr 2020 Online Presentations Contributor(s): Saaketh Desai
This tutorial introduces neural networks for classification tasks and random forests for regression tasks via Jupyter notebooks on nanoHUB.org. You will learn how to create and train a neural network to perform a classification, as well as how to define and train random forests. The tools used...

Hands-on Sequential Learning and Design of Experiments
29 Apr 2020 Online Presentations Contributor(s): Juan Carlos Verduzco Gastelum, Alejandro Strachan
This tutorial introduces the concept of sequential learning and information acquisition functions and how these algorithms can help reduce the number of experiments required to find an optimal candidate. A hands-on approach is presented to optimize the ionic conductivity of ceramic.