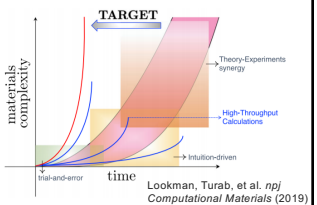
Data science and machine learning for material science

Saaketh Desai, Juan Carlos Verduzco, Ale Strachan
desai61@purdue.edu
jverduzc@purdue.edu
strachan@purdue.edu

School of Materials Engineering and Birck nanotechnology center
Purdue University

Materials by design



Lookman, Turab, et al. *npj Computational Materials* (2019)

Conve

Atomistic structure → Electronic structure → Physical parameters → Superionic? σ

~4 weeks/prediction

Our machine learning approach

Known structures + $\sigma_1, \sigma_2, ..., \sigma_n$ Known conductivities → Superionic? σ

Sendek, Austin D., et al. *Chemistry of Materials* (2018)

<1 second/prediction

50% F1 score to DFT results

Ferrium® C64®

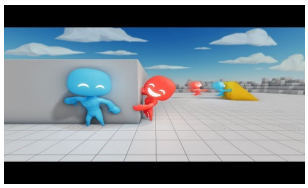
https://www.questek.com/ferrium-c64.html

2

Data science, machine learning and AI

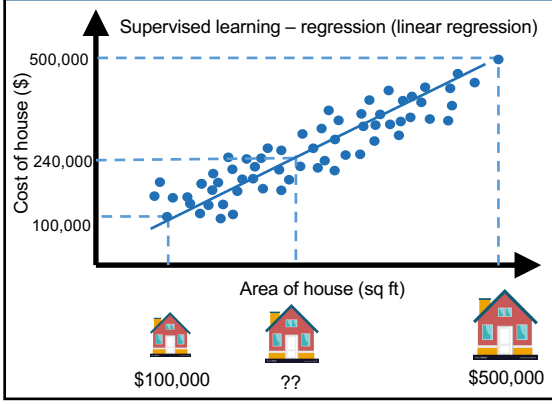
amazon

NETFLIX



3

Machine learning and data science

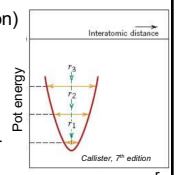


4

Linear regression to predict Young's modulus

Supervised learning – regression (linear regression)

- **Supervised learning:** Learning input-output relations based on a set of labeled 'ground truth' data
- **Regression:** Predicting a continuous relationship between inputs and outputs
- **Linear regression:** Assumed relationship to be linear



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

Introduction to Machine Learning for Materials Science

The tutorials here will give you an insight into the usage of Machine Learning to approach problems related to materials science.

- **Get started** Click on the links below to begin each tutorial.
- **Important** To exit individual tutorials and return to this page, use File -> Close and Halt. "Terminate Session" (top right) will kill your entire Jupyter session.

Querying databases, Organizing and Plotting Data:

- Query Pymatgen and Mendeleev for properties like Young's modulus and melting temperature
- Organize data into Pandas dataframes and python dictionaries and plot using Plotly

Linear Regression to predict material properties:

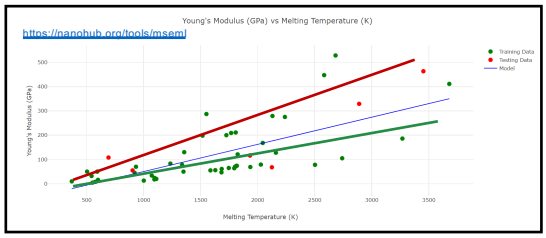
- Perform linear regression using the scikit learn package and predict Young's modulus
- Visualize trends in data and 'goodness of fit' of linear model

https://nanohub.org/tools/msemj

5

Obtaining good models – objective functions & gradient descent

https://nanohub.org/tools/msemj



- Objective function: metric to assess model quality
- Algorithm to direct model towards objective: Gradient descent

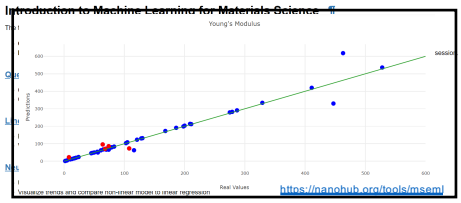
- Objective function: minimize sum of squared distances (points should lie as close to line as possible)
- Algorithm: move in the direction of largest decrease in error

6

Non-linear models: Neural networks

We could predict the Young's modulus given the melting temperature
Can we do better?

heat_of_formation	lattice_constant	melting_point	specific_heat	atomic_mass	atomic_radius	electrical_resistivity
284.9	4.09	1235.10	0.237	107.868200	1.60	1.630000e-08
330.9	4.05	933.50	0.900	26.981539	1.25	2.700000e-08
368.2	4.08	1337.58	0.129	196.966569	1.35	2.200000e-08
337.4	3.61	1356.60	0.385	63.546000	1.35	1.720000e-08
669.0	3.84	2683.00	0.133	192.217000	1.35	4.700000e-08



https://nanohub.org/tools/mseml

Neural networks – Forward propagation

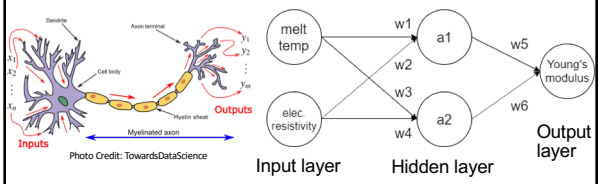


Photo Credit: TowardsDataScience

$$a_1 = f_1(w_1 melt.temp + w_2 elec.resistivity + b_1)$$

activation weight bias

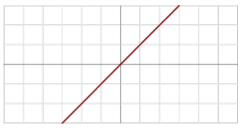
$$a_2 = f_2(w_3 melt.temp + w_4 elec.resistivity + b_2)$$

$$Young's\ modulus = f_3(w_5 a_1 + w_6 a_2 + b_3)$$

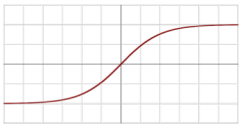
Training a neural network is an optimization problem where we solve for the weights and biases that result in accurate predictions

Neural networks – Activation functions

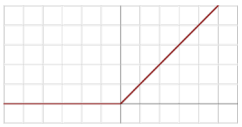
$$Linear$$
$$f(x) = x$$



$$Tanh$$
$$f(x) = \tanh(x)$$



$$Relu$$
$$f(x) = x\ if\ x > 0$$
$$f(x) = 0\ if\ x < 0$$



Neural networks – back propagation

$$a_1 = f_1(w_1 melt.temp + w_2 elec.resistivity + b_1)$$

activation weight bias

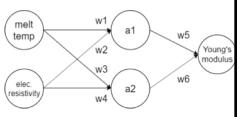
$$a_2 = f_2(w_3 melt.temp + w_4 elec.resistivity + b_2)$$

model prediction

$$\hat{y} = Young's\ modulus = f_3(w_5 a_1 + w_6 a_2 + b_3)$$

$$Objective = \frac{1}{N_{samples}} \sum_{i=1}^{N_{samples}} (\hat{y} - y)^2$$

ground truth



We need to update weights and biases such that objective function is minimized

$$w_1 = w_1 - \alpha \frac{\partial(Objective)}{\partial w_1}$$

Backpropagation

Weight update rule: gradient descent, Adam

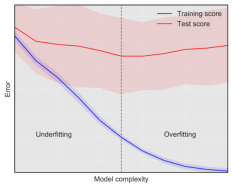
http://neuralnetworksanddeeplearning.com/chap2.html

Underfitting and overfitting

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

Image: Julien Despois

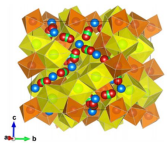


How do we train models that generalize well?

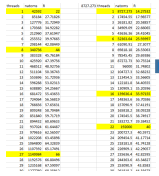
- Use a low enough learning rate
- Monitor error on validation set as a measure of model's ability to generalize

Sequential learning with the Citrination API

https://nanohub.org/tools/mldev Juan Carlos Verduzco



Rettenwander, Daniel, et al. *Chemistry of materials* 26.8 (2014)



https://citrination.com/users/sign_in

1. Querying a Database from Citrination

Matminer offers API tools to facilitate querying of databases like the Materials Project and Citrination. An individual **Citrine Key** is required for the query command *CitrineDataRetrieval*.

Data is stored in a Pandas Dataframe and the list of possible properties to be queried can be consulted by setting the print_properties_options parameter to **True**.

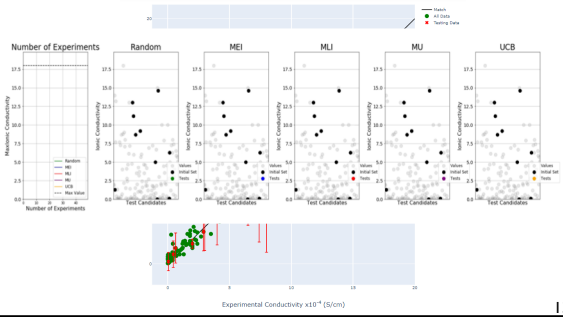
```
In [2]: cdr = CitrineDataRetrieval('B33skPqkMNhjmqOxZA9ADutt') # Citrine Key

        data = cdr.get_dataframe(criteria={'data_set_id': 184812}, print_properties_options=False) # LLZO Database
        property_interest = 'Ionic Conductivity' # Property to be queried

        display(data.head(n=10))
```

Consider making data publicly available to data platforms

Sequential learning with the Citrination API

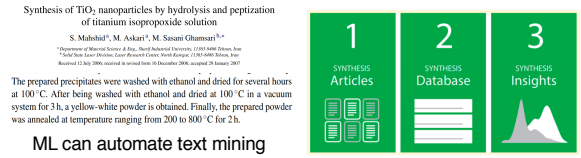
https://nanohub.org/tools/mldev Juan Carlos Verduzco

Sequential learning accelerates identification of high ionic conductivity garnets



13

Working with small datasets – other ML techniques

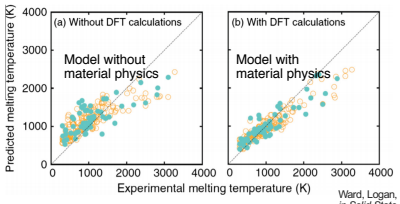

Synthesis of TiO_2 nanoparticles by hydrolysis and peptization of titanium isopropoxide solution

S. Mahshid[1], M. Askari[2], M. Sasani Ghamsari[3,*]

The prepared precipitates were washed with ethanol and dried for several hours at 100 °C. After being washed with ethanol and dried at 100 °C in a vacuum system for 3 h, a yellow-white powder is obtained. Finally, the prepared powder was annealed at temperature ranging from 200 to 800 °C for 2 h.

ML can automate text mining and information acquisition

Kim, Edward, et al. *Chemistry of Materials* (2017)

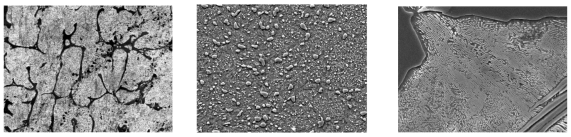


Incorporating physics into the model can significantly improve training even more smaller datasets!

Ward, Logan, and Chris Wolverton, *Current Opinion in Solid State and Materials Science* (2017)

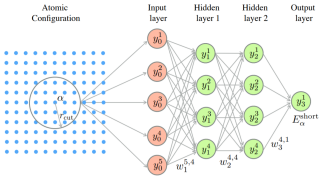
14

Other applications of neural networks



Ling, Julia, et al., *Materials Discovery* 10 (2017): 19-28

Convolutional neural networks can detect microstructural features



Artifical neural network interatomic potentials allow molecular dynamics simulations with almost quantum mechanical accuracy

Wen, Mingjian, and Ellad B. Tadmor, *arXiv preprint arXiv:1909.10134* (2019)

15

Obtaining and querying data

- Materials project: https://materialsproject.org/
- AFLOW: http://aflowlib.org/
- NIST Materials Data Repository: https://materialsdata.nist.gov/
- Database for renewable energy materials: https://materials.nrel.gov/
- OQMD: http://oqmd.org/

- Citrination: https://citrination.com/datasets
- ICSD: https://icsd.fiz-karlsruhe.de/

Databases with Python APIs:

- Pymatgen: https://pymatgen.org/
- Mendeleev: https://mendeleev.readthedocs.io/en/stable/data.html

- NanoHUB: https://nanohub.org/tools/mseml

16

Training neural networks

- Keras: https://materialsproject.org/
- Scikit-learn: https://scikit-learn.org/stable/
- PyTorch: https://pytorch.org/

- Matminer: https://hackingmaterials.lbl.gov/matminer/

- NanoHUB: https://nanohub.org/tools/mseml

17