Practical Introduction to Neural Network Potentials Day 1:

Potential energy surfaces, machine learning, & chemical data

The GitHub will contain all weekly exercises and slides:
https://github.com/zachglick/neural-network-potential-workshop

- You may come to as many or as few as you like
- The curriculum is subject to change
- Email us your suggestions and questions:
  - dmetcalf8@gatech.edu
  - zlg@gatech.edu

Potential energy

$$\hat{\mathrm{H}}|\Psi\rangle = E|\Psi\rangle$$

Can compute all of the observables from this,
but hard to solve, hence quantum chemistry as a field

Potential energy

$$\hat{\mathrm{H}}|\Psi\rangle = E|\Psi\rangle$$

Can compute all of the observables from this,
but hard to solve, hence quantum chemistry as a field

Separating the coupling of nuclear (**R**) and electronic coordinates (**r**), one can solve the simpler electronic equation:

$$\hat{\mathrm{H}}_e(\mathbf{r};\mathbf{R}) = \hat{\mathrm{T}}_e(\mathbf{r}) + \hat{\mathrm{V}}_{eN}(\mathbf{r};\mathbf{R}) + \hat{\mathrm{V}}_{ee}(\mathbf{r})$$

The Born-Oppenheimer approximation

$$\hat{\mathrm{H}}_e(\mathbf{r};\mathbf{R})|\Psi\rangle = E_e|\Psi\rangle$$

Potential energy

$$\hat{\mathrm{H}}|\Psi\rangle = E|\Psi\rangle$$

Can compute all of the observables from this,
but hard to solve, hence quantum chemistry as a field

Separating the coupling of nuclear (**R**) and electronic coordinates (**r**), one can solve the simpler electronic equation:

$$\hat{\mathrm{H}}_e(\mathbf{r};\mathbf{R}) = \hat{\mathrm{T}}_e(\mathbf{r}) + \hat{\mathrm{V}}_{eN}(\mathbf{r};\mathbf{R}) + \hat{\mathrm{V}}_{ee}(\mathbf{r})$$

$$\hat{\mathrm{H}}_e(\mathbf{r};\mathbf{R})|\Psi\rangle = \boxed{E_e}|\Psi\rangle$$

That's the stuff:
The energy is a function only of the nuclear coordinates

Potential energy

{Specific 3-dimensional coordinates of all nuclei}

```
0 1
C          -4.58735        0.92696        0.00000
C          -3.11050        0.92696        0.00000
H          -4.93786        1.78883        0.58064
H          -4.93786       -0.00682        0.45608
H          -4.93786        0.99888       -1.03672
H          -2.75999        0.85505        1.03672
H          -2.75998        1.86075       -0.45608
H          -2.75998        0.06509       -0.58064
```

{Quantum Chemistry}

E_e , or related quantities

- **DFT**
- **Hartree Fock**
- **Force fields**
- **MP2**
- **Coupled cluster**

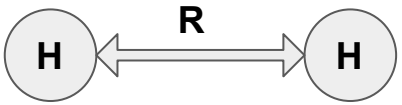
…

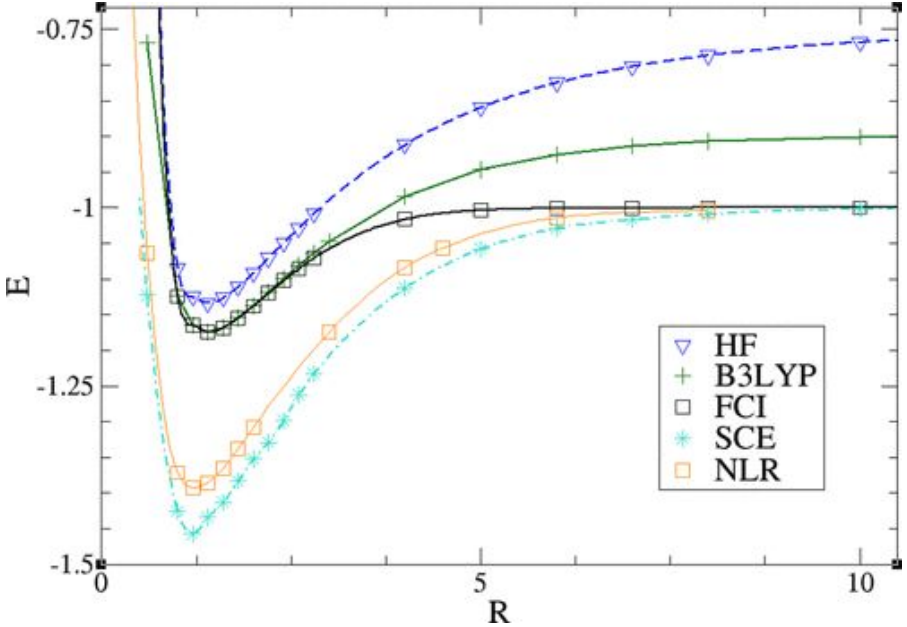
Potential energy surfaces (PES)

The potential energy over all possible coordinates of a system

Potential energy surfaces (PES)

The potential energy over all possible coordinates of a system

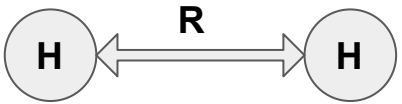




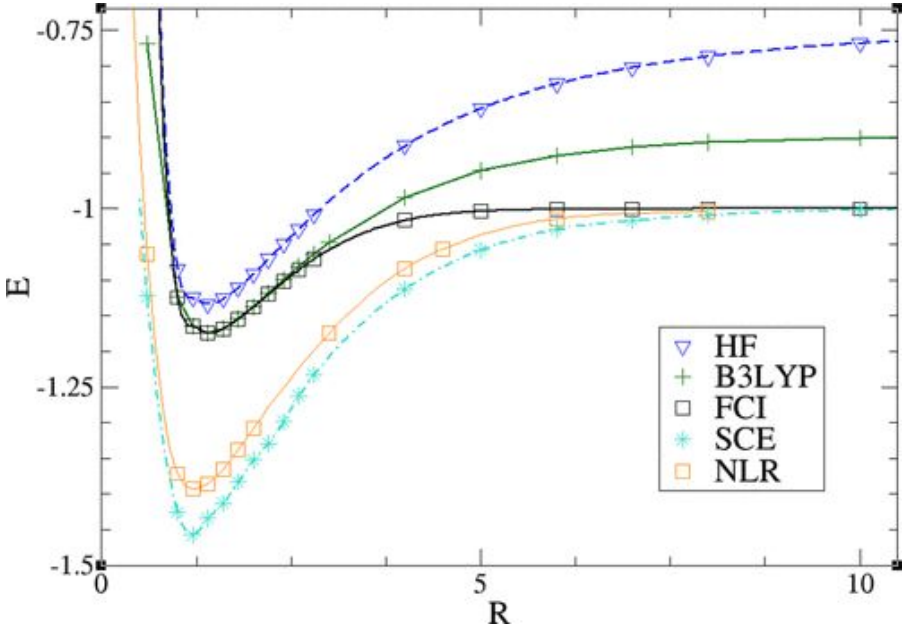
J. Chem. Theory Comput. 2015, 11, 7, 3153-3162

Potential energy surfaces (PES)

The potential energy over all possible coordinates of a system

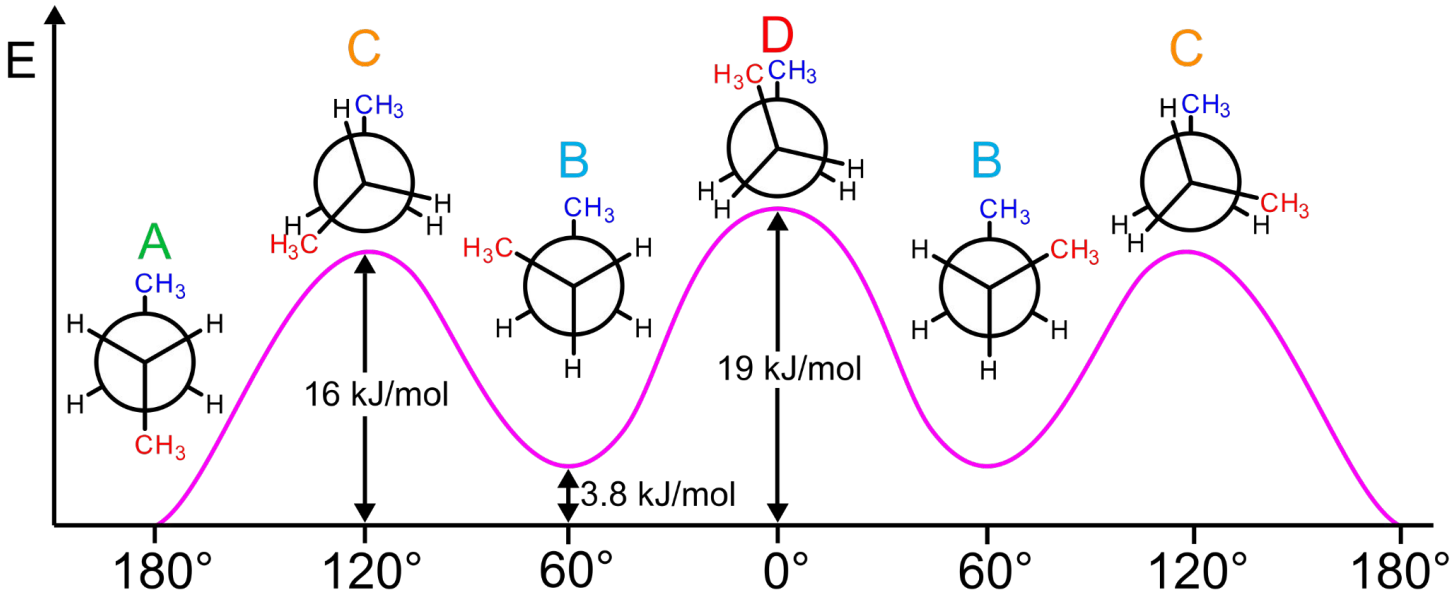


PES of a diatomic dissociation has *one* dimension,
most interesting systems are high dimensional
(≈3N-6 for N atoms)



J. Chem. Theory Comput. 2015, 11, 7, 3153-3162

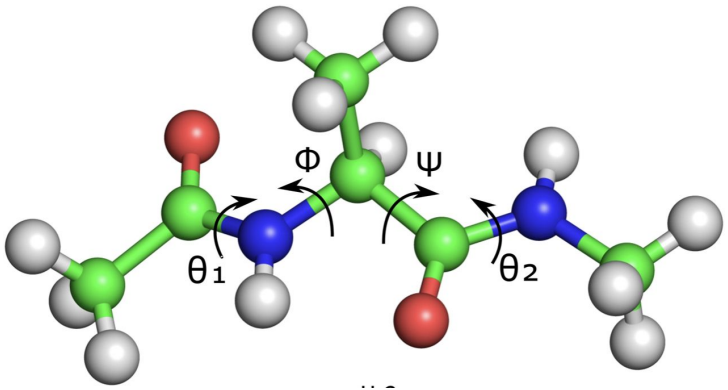
PES Applications: Understanding Isomerism

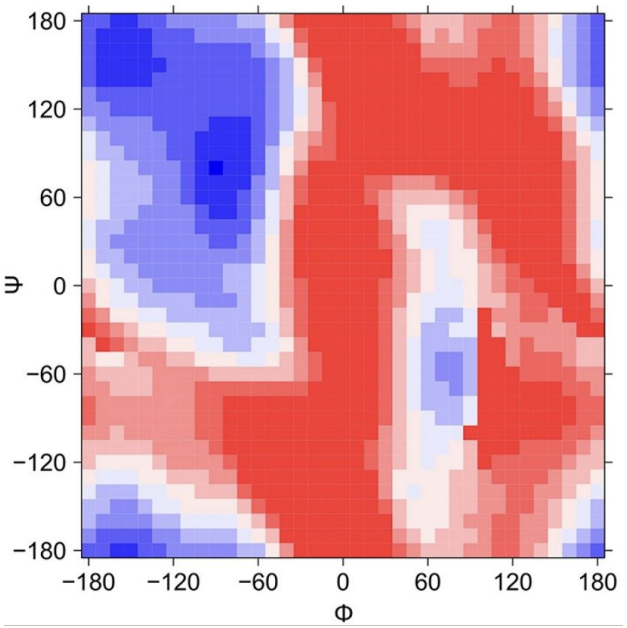


https://en.wikipedia.org/wiki/Molecular_dynamics

PES Applications: Stability of Biomolecules

Alanine dipeptide Ramachandran plot





J. Comput. Chem. 2019, 40, 297–309

PES Applications: Finding Transition States & Minima

Figure 1: Example example PES from C.D. Sherrill

Figure 1: Example PES from Steinfeld, Francisco, and Hase

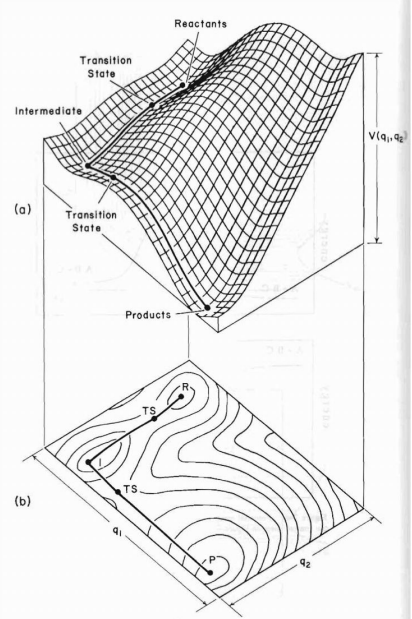
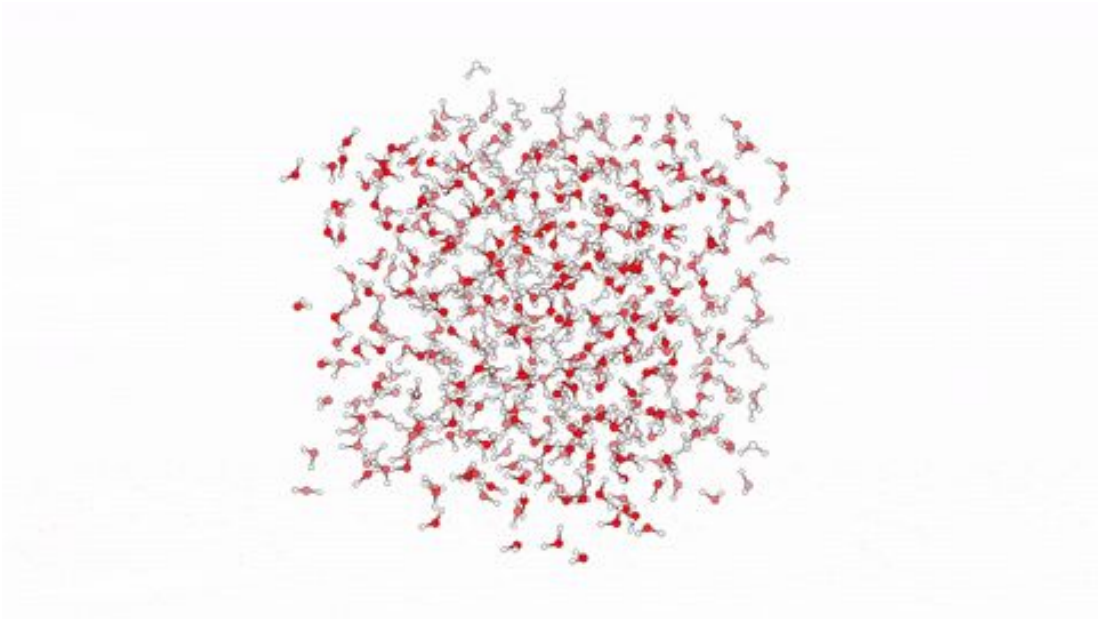


Figure 7–7. A three-dimensional perspective (a) and contour map (b) for a model chemical reaction. The solid line is the reaction path. [Adapted from G. M. Maggiora and R. E. Christoffersen, in *Transition States of Biochemical Processes*, ed. R. D. Gandour and R. L. Schowen (New York: Plenum, 1978).]

CDS notes on potential energy surfaces; accessed 2022

PES Applications: Molecular dynamics (MD)



https://en.wikipedia.org/wiki/Molecular_dynamics

What is machine learning (ML)?

Here, by ML we mean "performing non-linear regression"

$$x \mapsto y$$

Given many examples of (x, y),
find a function that approximates y of a new x.

What is machine learning (ML)?

Here, by ML we mean "performing non-linear regression"

$$x \mapsto y$$

Model: $$\hat{y} = f(x; \theta)$$

Model parameters

What is machine learning (ML)?

Here, by ML we mean "performing non-linear regression"

$$x \mapsto y$$

Model: $$\hat{y} = f(x; \theta)$$

Model parameters

$$\theta = \arg\min \underbrace{(y - \hat{y})^2}_{\text{Error measurement, "loss"}}$$

What is machine learning (ML)?

Here, by ML we mean "performing non-linear regression"

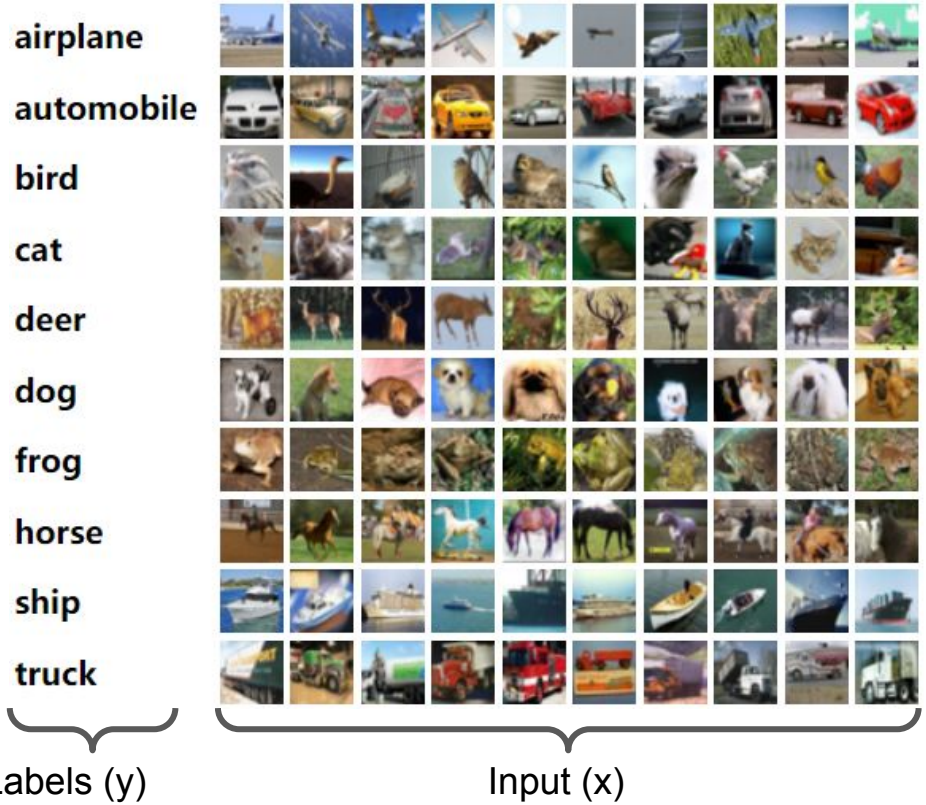
$$x \mapsto y$$

Model: $$\hat{y} = f(x; \theta)$$

$$\theta = \arg\min \, (y - \hat{y})^2$$

Given examples of (x, y), what are the optimal parameters θ?

Image $\mapsto$ English word



CIFAR 10 dataset e.g.

ML needs data, what is there for energies (and related)?

$$\mathbf{R} \mapsto E$$

Unlike other fields, data can be generated computationally:

- QM9 enumerates simple organic molecules with 9 or fewer heavy atoms and their QM-computed energies
- MD-17 consists of MD trajectories of 8 organic molecules along with QM-computed energies and forces
- ANI-1 (1x, ccx, 2) datasets …

Most datasets are small molecule geometries (xyz) and some target energy or force from QM.

https://qcarchive.molssi.org/apps/ml_datasets/ : The Molecular Sciences Software Institute dataset repository

Modeling energies

Mapping: $$x \mapsto y$$

Model: $$\hat{y} = f(x; \theta)$$

$$\mathbf{R} \mapsto E$$

$$\hat{E} = \boxed{f(\mathbf{R}; \theta)}$$

What form should this function take?

Modeling energies

$$\mathbf{R} \mapsto E$$

$$\hat{E} = \boxed{f(\mathbf{R}; \theta)}$$

There are many, many choices.
Most are bad, do not pick randomly.

1. Supervised learning

1.1. Linear Models
- 1.1.1. Ordinary Least Squares
- 1.1.2. Ridge regression and classification
- 1.1.3. Lasso
- 1.1.4. Multi-task Lasso
- 1.1.5. Elastic-Net
- 1.1.6. Multi-task Elastic-Net
- 1.1.7. Least Angle Regression
- 1.1.8. LARS Lasso
- 1.1.9. Orthogonal Matching Pursuit (OMP)
- 1.1.10. Bayesian Regression
- 1.1.11. Logistic regression
- 1.1.12. Generalized Linear Regression
- 1.1.13. Stochastic Gradient Descent - SGD
- 1.1.14. Perceptron
- 1.1.15. Passive Aggressive Algorithms
- 1.1.16. Robustness regression: outliers and modeling errors
- 1.1.17. Quantile Regression
- 1.1.18. Polynomial regression: extending linear models with basis functions

1.2. Linear and Quadratic Discriminant Analysis
- 1.2.1. Dimensionality reduction using Linear Discriminant Analysis
- 1.2.2. Mathematical formulation of the LDA and QDA classifiers
- 1.2.3. Mathematical formulation of LDA dimensionality reduction
- 1.2.4. Shrinkage and Covariance Estimator
- 1.2.5. Estimation algorithms

1.3. Kernel ridge regression

1.4. Support Vector Machines
- 1.4.1. Classification
- 1.4.2. Regression
- 1.4.3. Density estimation, novelty detection
- 1.4.4. Complexity
- 1.4.5. Tips on Practical Use
- 1.4.6. Kernel functions
- 1.4.7. Mathematical formulation
- 1.4.8. Implementation details

1.5. Stochastic Gradient Descent
- 1.5.1. Classification
- 1.5.2. Regression
- 1.5.3. Online One-Class SVM
- 1.5.4. Stochastic Gradient Descent for sparse data
- 1.5.5. Complexity
- 1.5.6. Stopping criterion
- 1.5.7. Tips on Practical Use
- 1.5.8. Mathematical formulation
- 1.5.9. Implementation details

1.6. Nearest Neighbors
- 1.6.1. Unsupervised Nearest Neighbors
- 1.6.2. Nearest Neighbors Classification
- 1.6.3. Nearest Neighbors Regression
- 1.6.4. Nearest Neighbor Algorithms
- 1.6.5. Nearest Centroid Classifier
- 1.6.6. Nearest Neighbors Transformer
- 1.6.7. Neighborhood Components Analysis

1.7. Gaussian Processes
- 1.7.1. Gaussian Process Regression (GPR)
- 1.7.2. GPR examples
- 1.7.3. Gaussian Process Classification (GPC)
- 1.7.4. GPC examples
- 1.7.5. Kernels for Gaussian Processes

1.8. Cross decomposition
- 1.8.1. PLSCanonical
- 1.8.2. PLSSVD
- 1.8.3. PLSRegression
- 1.8.4. Canonical Correlation Analysis

1.9. Naive Bayes
- 1.9.1. Gaussian Naive Bayes
- 1.9.2. Multinomial Naive Bayes
- 1.9.3. Complement Naive Bayes
- 1.9.4. Bernoulli Naive Bayes
- 1.9.5. Categorical Naive Bayes
- 1.9.6. Out-of-core naive Bayes model fitting

1.10. Decision Trees
- 1.10.1. Classification
- 1.10.2. Regression
- 1.10.3. Multi-output problems
- 1.10.4. Complexity
- 1.10.5. Tips on practical use
- 1.10.6. Tree algorithms: ID3, C4.5, C5.0 and CART
- 1.10.7. Mathematical formulation
- 1.10.8. Minimal Cost-Complexity Pruning

1.11. Ensemble methods
- 1.11.1. Bagging meta-estimator
- 1.11.2. Forests of randomized trees
- 1.11.3. AdaBoost
- 1.11.4. Gradient Tree Boosting
- 1.11.5. Histogram-Based Gradient Boosting
- 1.11.6. Voting Classifier
- 1.11.7. Voting Regressor
- 1.11.8. Stacked generalization

1.12. Multiclass and multioutput algorithms
- 1.12.1. Multiclass classification
- 1.12.2. Multilabel classification
- 1.12.3. Multiclass-multioutput classification
- 1.12.4. Multioutput regression

1.13. Feature selection
- 1.13.1. Removing features with low variance
- 1.13.2. Univariate feature selection
- 1.13.3. Recursive feature elimination
- 1.13.4. Feature selection using SelectFromModel
- 1.13.5. Sequential Feature Selection
- 1.13.6. Feature selection as part of a pipeline

1.14. Semi-supervised learning
- 1.14.1. Self Training
- 1.14.2. Label Propagation

Regression models from popular Python library scikit-learn

Model Selection

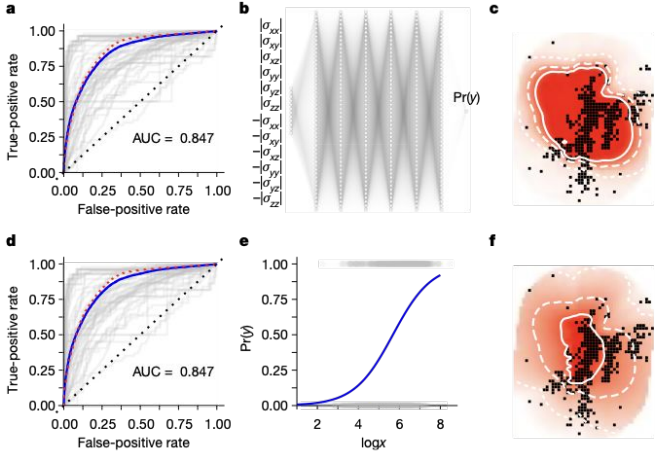
A *carefully designed* two-parameter model outperforms a 13K parameter deep neural network

MATTERS ARISING

https://doi.org/10.1038/s41586-019-1582-8

One neuron versus deep learning in aftershock prediction

Arnaud Mignan[1,2,3]* & Marco Broccardo[2,4]*



ıftershock spatial patterns based on stress perating characteristic (ROC) curves of test data

e, Logistic regression fit on training data. **f,** Example of prediction. For both DNN and logistic regression mode

Exercise #1
Modeling Energies with Linear Regression

Modeling Energies with Linear Regression

Let’s start simple; add complexity only when needed.

Why linear regression?

1. Simplest possible model
2. We know that E is **extensive** (grows with system size)

$$\hat{y} = \vec{w} \cdot \vec{x} + b$$

$$\hat{y} = \sum_i w_i \cdot x_i + b$$

Modeling Energies with Linear Regression

The length of $\hat{x}$ must be the same for all molecules

Can’t (and shouldn’t) use raw coordinates

```
3

O            0.000000000000     0.000000000000    -0.064747837931
H           -0.000000000000    -0.748813239898     0.513797588642
H            0.000000000000     0.748813239898     0.513797585505
```

```
5

Cl           0.558980060028    -0.000003409116    -0.000010838753
C           -1.233013253888     0.000007518693     0.000023805641
H           -1.571271543066    -0.697791943059     0.759266356426
H           -1.571271410823     1.006427246976     0.224727628339
H           -1.571303371347    -0.308606540429    -0.983901358501
```

Modeling Energies with Linear Regression

The length of $\hat{x}$ must be the same for all molecules

How about atom counts?

	N_H	N_C	N_O	N_{Cl}
3 O H H	2	0	1	0
5 Cl C H H H	3	1	0	1

Modeling Energies with Linear Regression

The length of $\hat{x}$ must be the same for all molecules

How about atom counts?

$$\sum_{i}^{N_{elem}} w_i \cdot N_i + b$$

	N_H	N_C	N_O	N_{Cl}
3 O H H	**2**	**0**	**1**	**0**
5 Cl C H H H	**3**	**1**	**0**	**1**

$$\hat{E} = 2w_H + 1w_O + b$$

$$\hat{E} = 3w_H + 1w_C + 1w_{Cl} + b$$

Modeling Energies with Linear Regression

Workshop repo: https://github.com/zachglick/neural-network-potential-workshop

Exercise #1

- Analyze the distribution of energies* in the QM9 dataset
- Fit a simple linear regression model to predict **energy** from **atom counts**

*Ground state DFT energies at optimized geometries