

CyberMAGICS Workshop: Introduction to Machine Learning

Ken-ichi Nomura

*Collaboratory for Advanced Computing & Simulations
University of Southern California*

ML software hands-on: Ankit Mishra, Nitish Baradwaj,
Ruru Ma



**Supported by National Science
Foundation, Award OAC-2118061**

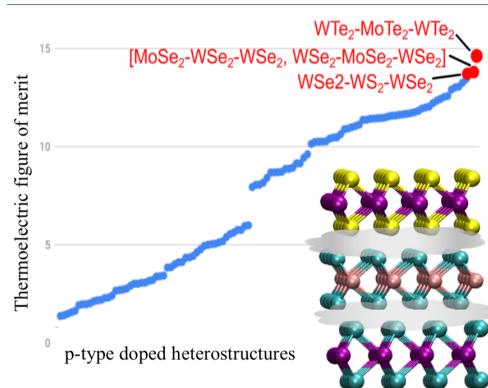


CyberMAGICS Workshop, June 30, 2022

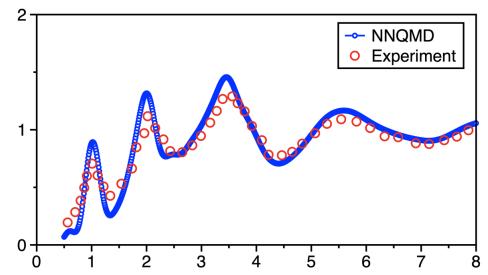
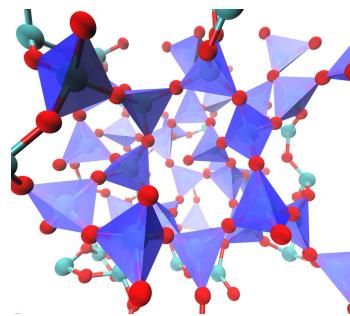
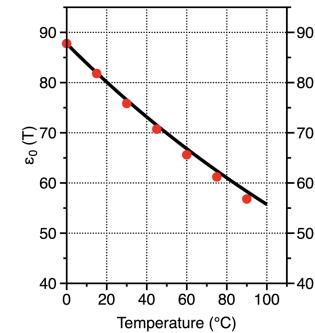
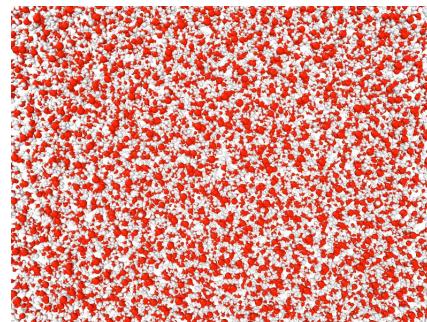


Material Modeling with Machine Learning

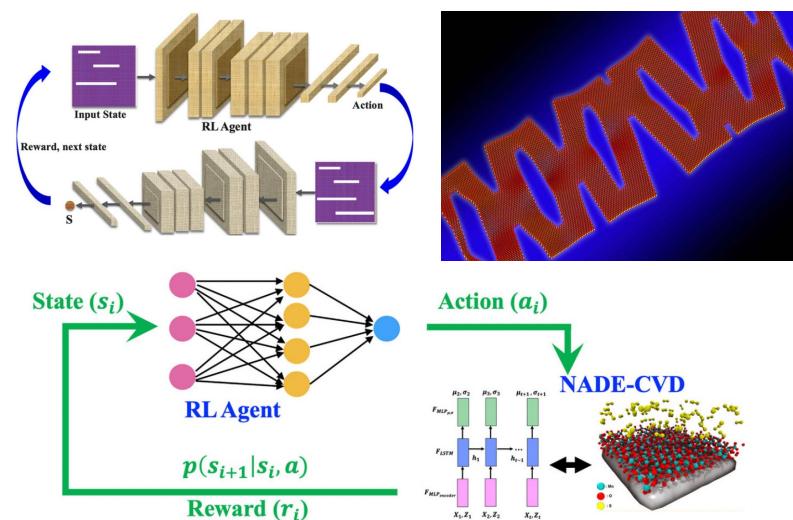
Active learning for accelerated material design



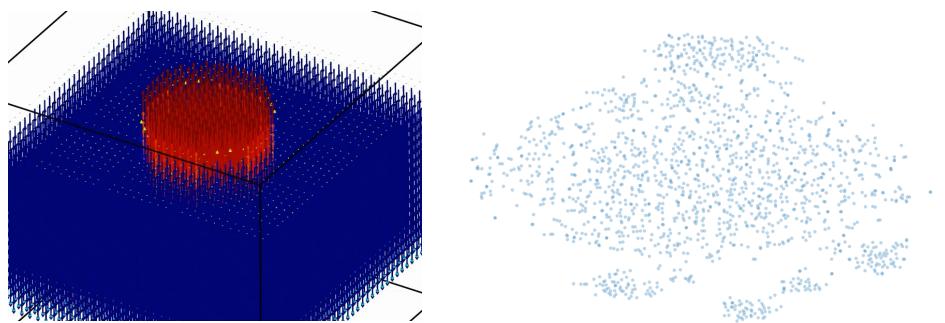
Large-scale and long-time neural network QMD simulations



Reinforcement learning for quantum materials synthesis

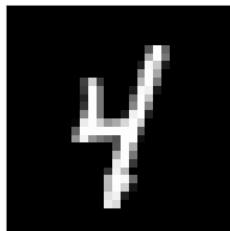


Deep generative model for ferroelectrics

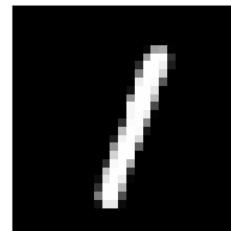


What is Machine Learning?

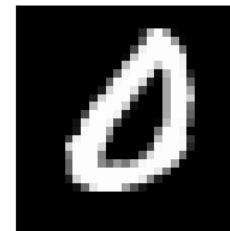
Image classification using MNIST dataset



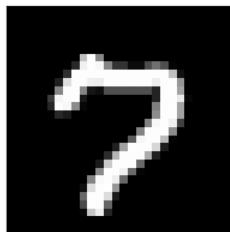
4 (4)



1 (1)



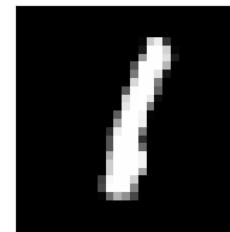
0 (0)



7 (7)



8 (8)



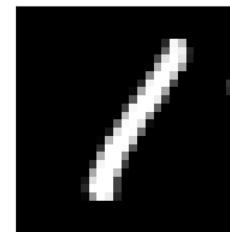
1 (1)



2 (2)



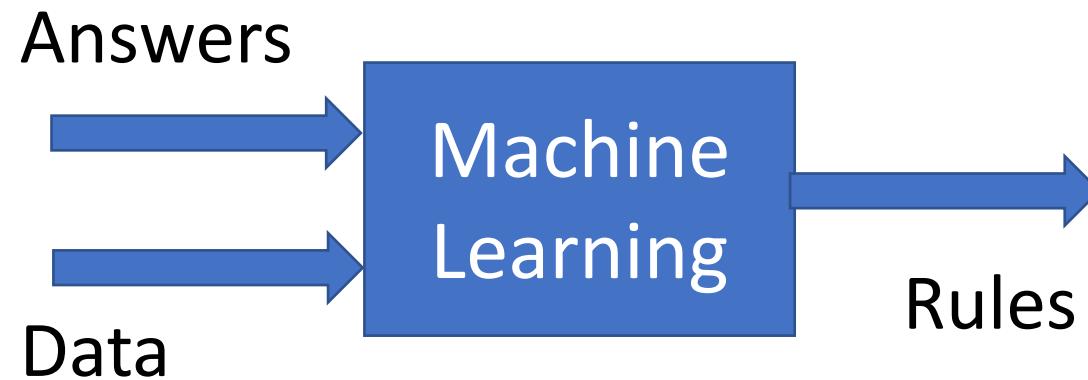
7 (7)



1 (1)

What is Machine Learning?

What is Machine Learning?



Classification vs Regression

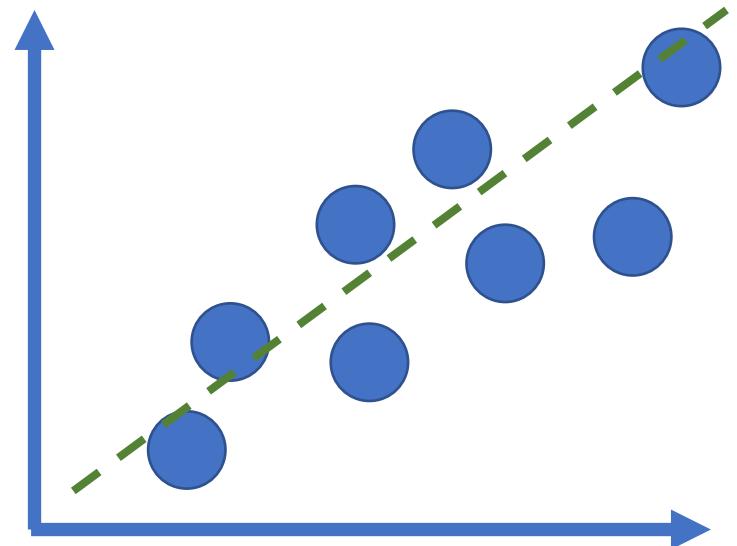
Classification

Predict class label



Regression

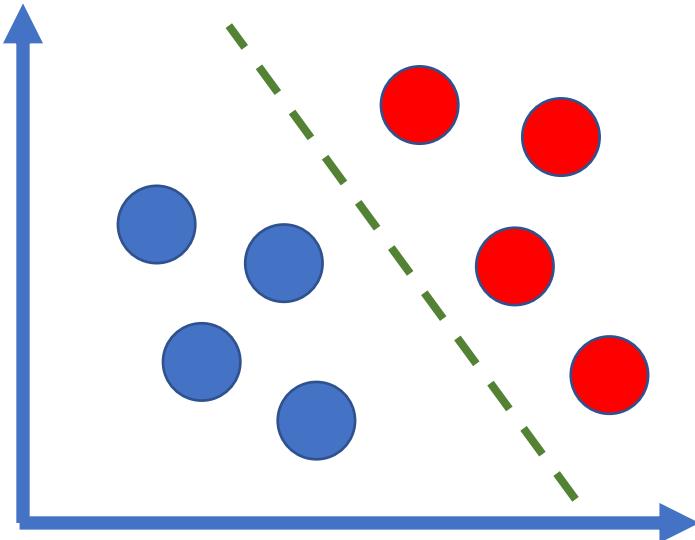
Predict real value



Supervised vs Unsupervised

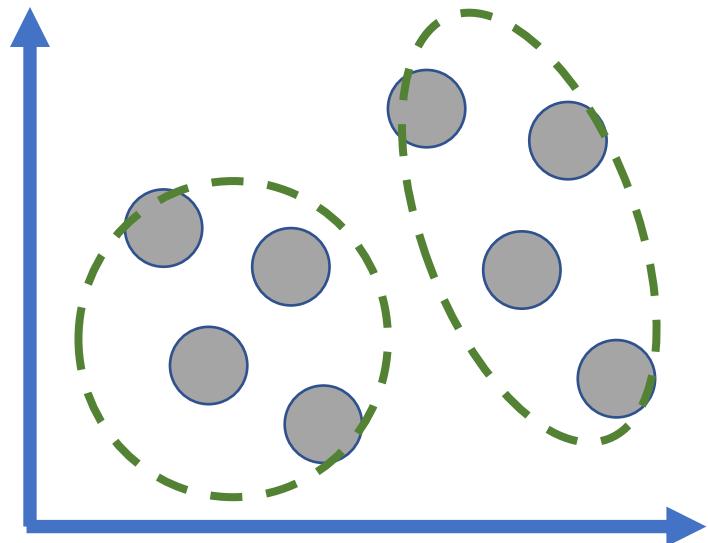
Supervised:

Data are "labeled"
classification, regression



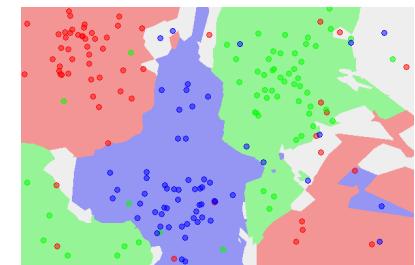
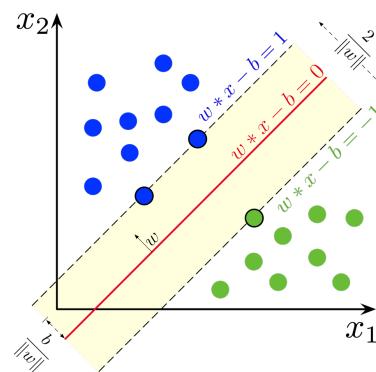
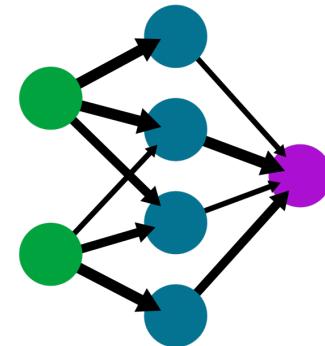
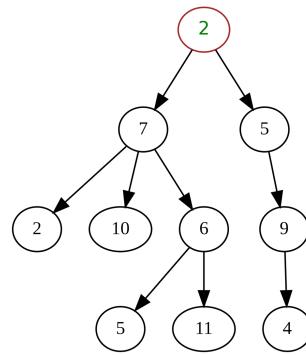
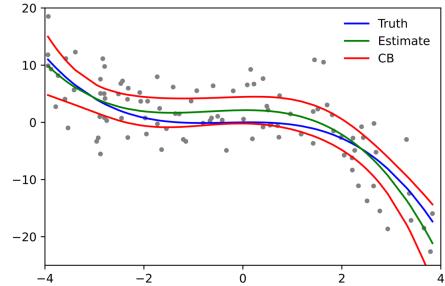
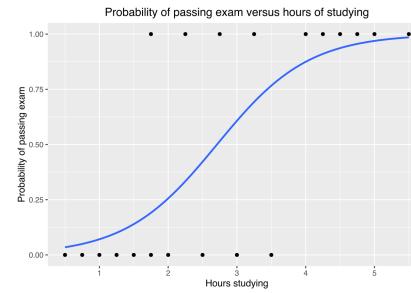
Unsupervised:

Data are “not labeled”
Clustering, dimensionality reduction



ML Algorithms

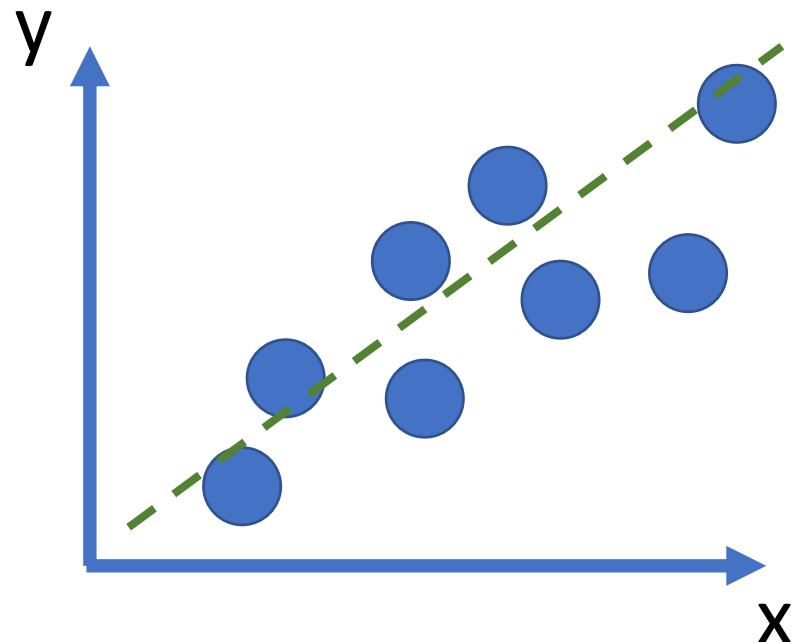
- Linear/polynomial Regressions
- Logistic Regression
- K-Nearest Neighbors
- Decision Trees
- Random Forests
- Support Vector Machines
- Neural Networks
- Bayesian Networks
- PCA & t-SNE



Linear Regression

- Assumes a linear relationship between the input variable(s) and the output variable (y)
- Can be univariate, multivariate, polynomial, logarithmic, ...
- Coefficients (b_i) are obtained by minimizing the sum of the difference between all data and line

$$\hat{y} = \beta_0 + \beta_1 x_1 + \cdots + \beta_n x_n = x_i^T \boldsymbol{\beta}$$



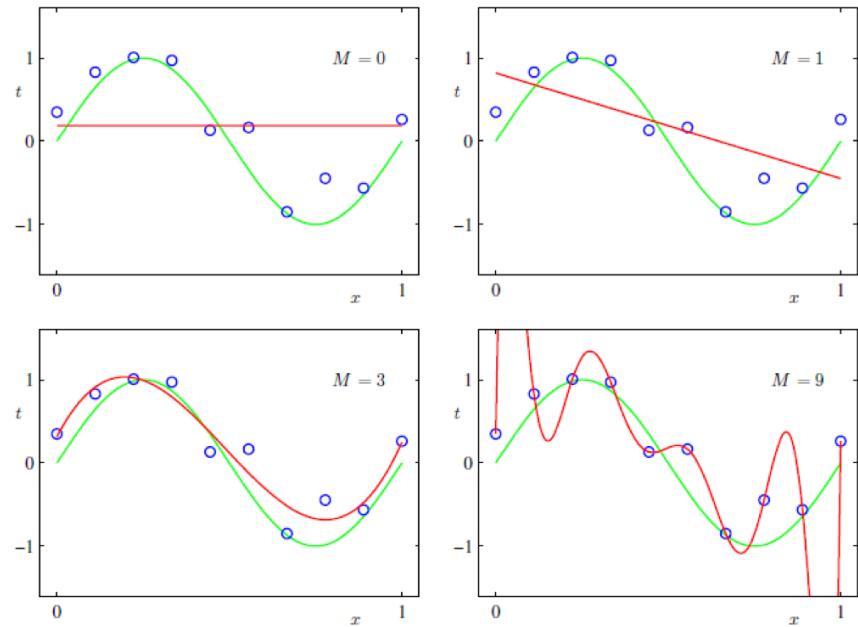
$$L = \frac{1}{N} \sum_{i=1}^N (\hat{Y}_i - Y_i)^2$$

Overfitting and Regularization

- A good ML model should accurately predict existing training data as well as “unseen” (out-of-sample) data
- A model with many parameters tends to pick up noise in data and poorly perform on unseen data, i.e. overfitting
- Regularizations, such as Ridge and LASSO

$$\min_{\beta, \beta_0} \left\{ \frac{1}{N} \|y - X\beta\|_2^2 \right\} \text{ with } \|\beta\|_1 \leq t \text{ or } \|\beta\|_2^2 \leq t$$

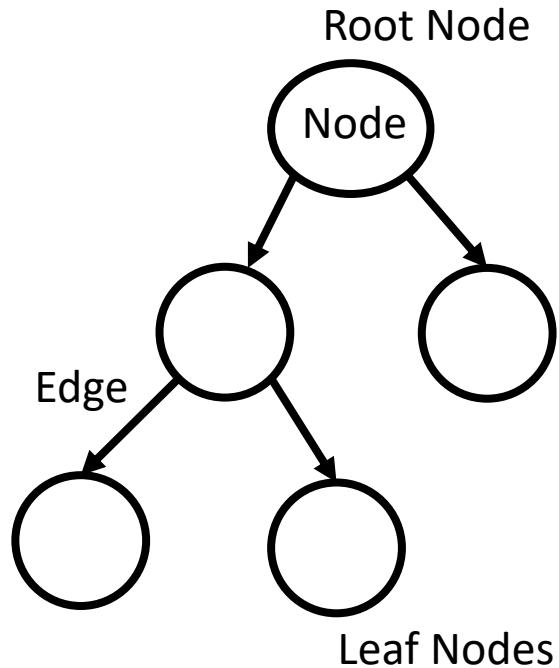
$$y = \beta_0 + \beta_1 x^1 + \cdots + \beta_M x^M$$



True function
Training data with noise
Model predictions

Decision Tree and Random Forest

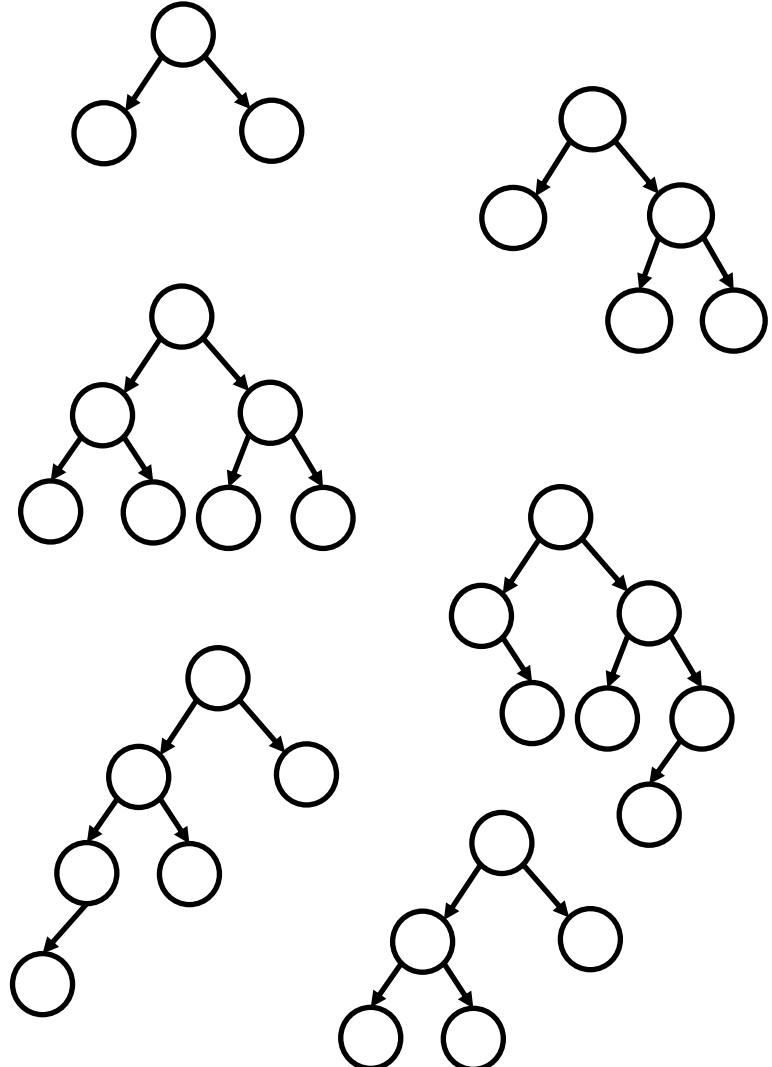
- Used for classification or regression
- Starting from root node, “ask a question and select an answer” until a leaf node is reached
- Tree construction based on information theory
 - Gini index/entropy for classification
 - Variance/RMSE for regression
- Easy to construct and interpret, but also overfit



$$I_{Gini} = 1 - \sum_j p_j^2$$
$$I_{entropy} = - \sum_j p_j \log(p_j)$$

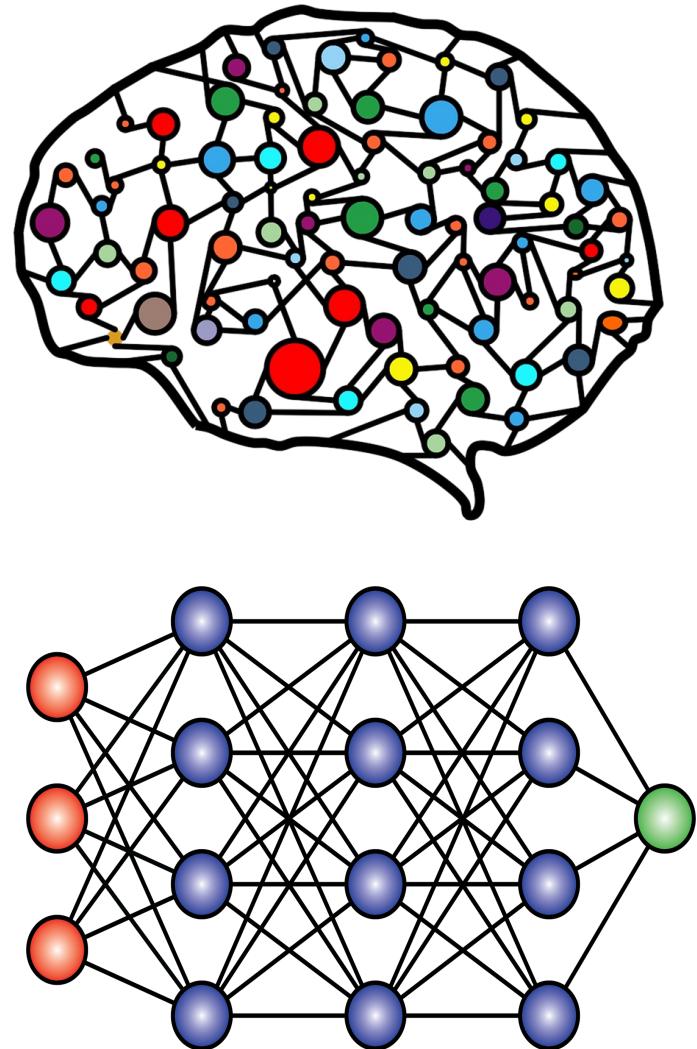
Decision Tree and Random Forest

- Ensemble of decision trees
- Aggregate predictions from each tree as the model prediction
- Good prediction accuracy, generalizability, robust to overfitting
- Less interpretability to single decision tree

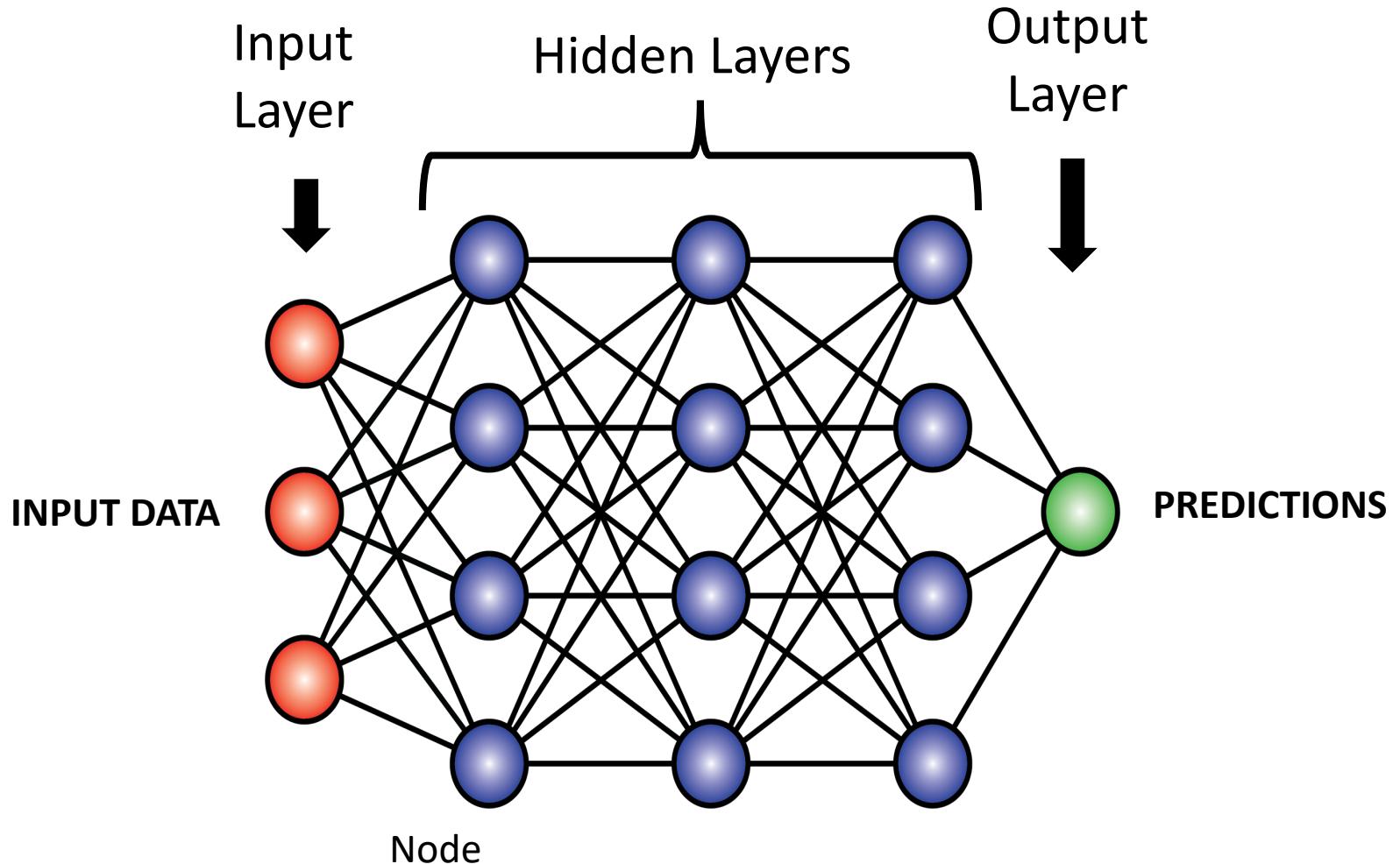


Neural Network

- Inspired by biological brain
- A universal function approximator
- A key component in other deep learning algorithms
- Hyperparameters
 - Number of nodes
 - Degree of connectivity of nodes
 - Number of layers in network

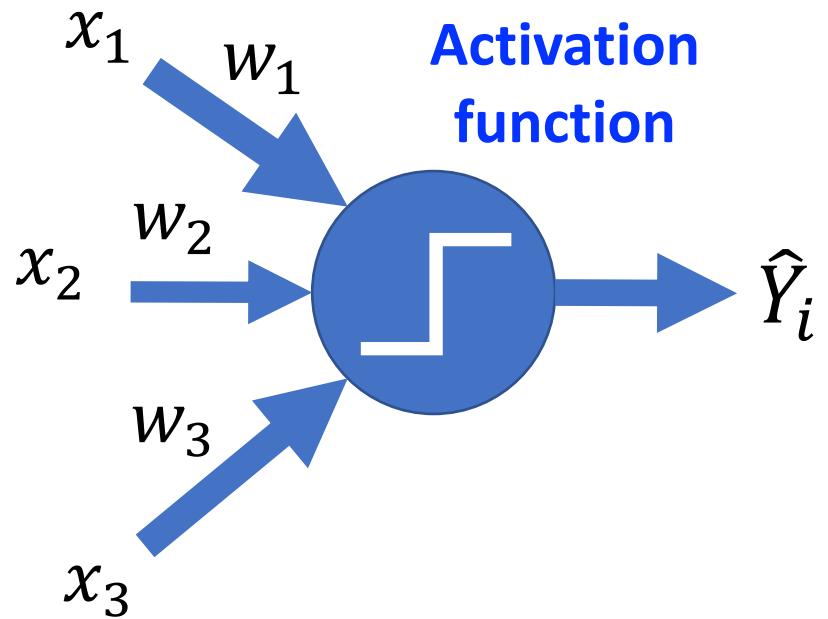


Neural Network



Neural Network

- On each node, outputs (x) from previous layer are aggregated with weights (w)
- A non-linear activation function transforms the aggregated inputs and pass it to next layer
- Compute Loss function (difference between model prediction and given true value) after the output layer

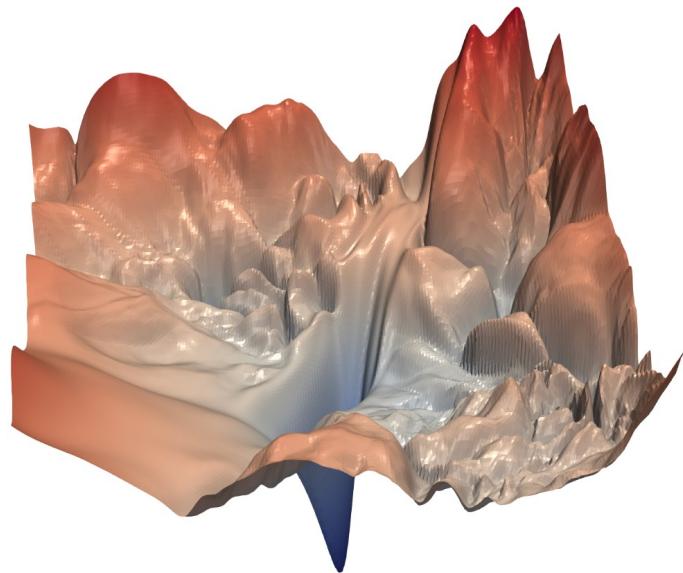
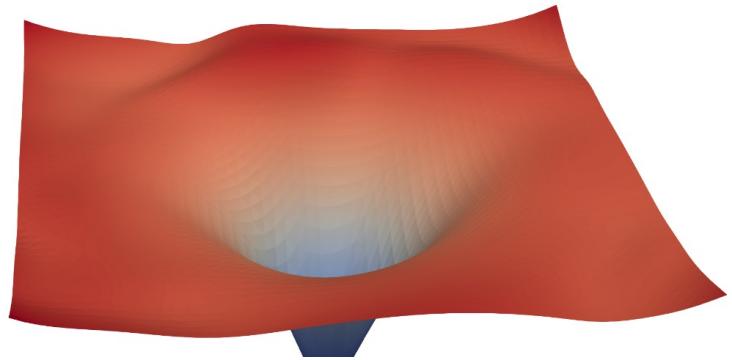
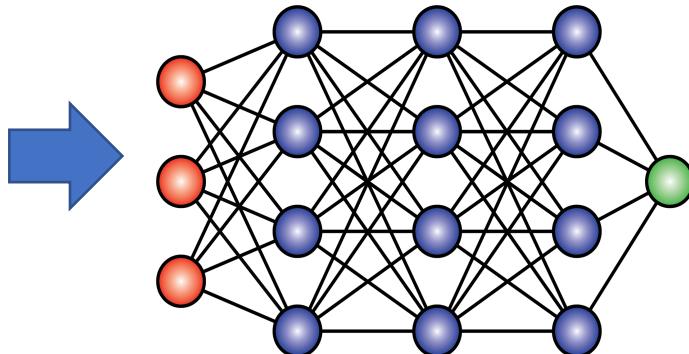


$$L = \frac{1}{N} \sum_{i=1}^N (\hat{Y}_i - Y_i)^2$$

Neural Network Training

- Network parameters are “trained” by minimizing loss function
- Stochastic gradient decent is commonly used

$$\Delta w = -\partial L / \partial w$$



Loss function landscape

Moving Forward

- **Linear algebra, Statistics & Probability, Python**

- **Online courses**

<https://www.coursera.org/browse/data-science/machine-learning>

- **Textbooks**

Deep Learning

Ian Goodfellow and Yoshua Bengio and Aaron Courville

<https://www.deeplearningbook.org/>

The Elements of Statistical Learning

Trevor Hastie, Robert Tibshirani, Jerome Friedman

<https://hastie.su.domains/ElemStatLearn/>

- **Python Programming**

Scikit-learn <https://scikit-learn.org>

Pytorch <https://pytorch.org>

Tensorflow <https://www.tensorflow.org>