Machine Learning for PHYSICS

2018. 02. 21 @ Korea University
HELLO!

I am Chang-Woo Lee
@ Korea Institute for Advanced Study

If you can have any *data* unwieldy to handle,
I would be happy to help you to train a machine for it!
Contents

- Introduction to Machine Learning (ML)
  - Data Mining vs. Machine Learning
  - Background Theories for ML
  - History of ML
  - Methodologies of ML
  - ML Models (Kinds of Machines)
  - Implementation
  - Learning about Neural Network

- Application of ML to Physics
“I visualize a time when we will be to robots what dogs are to humans, and I’m rooting for the machines.” — Claude Shannon
Introduction to Machine Learning

Sketchy overview
DM vs. ML

- **Data Mining (DM)**
  - Focus on discovering *patterns*, unknown properties & internal relations of data (e.g. fraud detection)

- **Machine Learning (ML)**
  - Focus on extracting *prediction* models based on learned (but mostly hidden) properties using training data (e.g. e-mail spam auto-classification)

- **Often Overlapped**
  - They employ each other as a tool.
  - Clustering, anomaly detection, ...
Two Paradigms of AI

- **Symbolism (Top-Down)**
  - represents information through *symbols* and their *relationships*
  - well-suited for representing *explicit* knowledge that can be appropriately formalized

- **Connectionism (Bottom-Up)**
  - Mental phenomena are *emergent* processes of interconnected networks of *simple* units.
  - Biological learning is mostly *implicit*—it is an adaptation process based on uncertain information and reasoning.

- **Hybrid model**
  - Low-level tasks ⇒ connectionist model
Background Theories

- **Universal Approximation Theorem**
  - A feed-forward network with a single hidden layer containing a finite number of neurons (i.e., a multilayer perceptron), can approximate continuous functions on compact subsets of $\mathbb{R}^n$.

- **Representer Theorem**
  - A minimizer $f^*$ of a regularized empirical risk function can be represented as a finite linear combination of kernel products evaluated on the input points in the training set data.

- **(Naïve) Bayesian Inference**
  - Assuming joint probability (& independent features)
    - $\Rightarrow$ $(\text{training data}) \propto$ not exponentially but linearly to $(\text{features})$

- **No Free Lunch Theorem**
  - Do not expect too much from single ML since there is no one model that works best for every problem.
Background Theories

- **Bias-Variance Tradeoff**
  - Bias: difference between a model's expected predictions and the true values
  - Variance: algorithm's sensitivity to specific sets of training data
Background Theories

- **Discriminative Models**
  - focus on (if any) $P(y|x)$ ($x$: data, $y$: class)
  - yield a (hard or soft) *boundary* between classes (and probably a more complex one)
  - require fewer assumptions
  - often perform better when #(data) is large and your generative assumptions do not satisfy

- **Generative Models**
  - try to learn on $P(x|y) & P(y)$, equivalently $P(x,y)$
  - model the *distribution* of individual classes
  - show good performance when #(data) is small or $\exists$ (missing data)
  - Generally, probabilistic graphical models offer rich representation.
**Adaline (Adaptive Linear Element)**

- Equivalent to perceptron except that Adaline uses delta/LMS rule, i.e. cost function
- Delta rule $\rightarrow$ back propagation
Multilayer Perceptron

- **XOR Problem**
  - Single-layer perceptron cannot solve XOR classification
  - The First AI Winter (1969)
  - ⇒ Multilayer perceptron

- **Increasing Computation**
  - ⇒ Back propagation, CNN

- **Vanishing (or exploding) Gradient Problem**
  - The Second AI Winter (1990’s)
  - SVM
  - Solution ⇒ LSTM, ReLU, Batch Normalization, ...
Resurgence of Deep Neural Net

- Huge Computational Cost
  - Exponentially increasing # of weights
  - Unsupervised pre-training: Autoencoder, DBN
  - GPGPU (General-purpose GPU), mini-batch

- Huge Number of Data Required
  - Era of big data

- Overfitting
  - Cross-validation, dropout, regularization, ensembling (bagging/boosting)
Supervised Learning

- Classification & Regression
  - Needs *labeled* data
  - Given \( \{x_i, y_i\} \), learn \( y = f(x; \theta) \)
  - Classification: \( y \) is categorical, e.g. digit recognition
  - Regression: \( y \) is continuous, e.g. temperature, stock price
MNIST Training
Unsupervised Learning

- **Clustering, Anomaly Detection**
  - *Unlabeled* data $\Rightarrow$ clustering, anomaly detection
  - Given $\{x_i\}$, learn $y=f(x; \theta)$

SVM: Support Vector Machine, KNN: k-Nearest Neighbors, CART: Classification and Regression Tree
LASSO: Least Absolute Shrinkage and Selection Operator, PCA: Principal Component Analysis, ICA: Independent Component Analysis
“Science is the systematic classification of experience.”
— George Henry Lewes
Reinforcement Learning

Markov Decision Process (MDP)

- Extension of Markov chain (MC): If rewards are the same, MDP $\Rightarrow$ MC
- 5-tuple $(S, A, P(s, s'), R(s, s'), \gamma)$
- Goal: find a policy $\pi$ that will maximize sum of rewards $\sum_{t=0}^{\infty} \gamma^t R_{a_t}(s_t, s_{t+1})$
- focuses on performance; try to balance between exploration (of uncharted territory) and exploitation (of current knowledge) $\Rightarrow$ exploration-exploitation trade-off

Q-Learning

- $Q(s, a)$: action-value function
- $Q$ learned $\Rightarrow \pi$ (just follow max $Q$)
- No explicit specification of the transition probabilities
- Theorem: For any finite MDP, Q-learning eventually finds an optimal policy (i.e. maximum reward is achievable).

NIPS Deep Learning Workshop 2013
NEURAL NETWORK ZOO
http://www.asimovinstitute.org/neural-network-zoo/

✓ A variety of types of neural networks available depending on tasks
✓ Frequently used models in physics: DFF, RBM/DBN, CNN, SVM, RBF, ...
   (many other machines are waiting to be used!)
Feed-Forward NN

One of Basic NNs

- Fully connected layers
- Layers: Input + hidden + output
- Mostly good choice when #(labeled data) is large

http://playground.tensorflow.org
Compressed Representation of Input Data

- Unsupervised generative learning of $f(x) \approx x$
- denoising/sparse/variational AE

Autoencoder

http://playground.tensorflow.org
Convolutional Neural Network

- Specialized for Image Recognition
  - Inspired by biological process of visual cortex
  - Convolution \(\Rightarrow\) filtering
  - Shift-invariant
  - Good for feature extraction
  - Layers: convolution, pooling, fully-connected
Restricted Boltzmann Machine

- **Bipartite-graph connections**
  - Probabilistic model
  - Usually trained with contrastive divergence (CD) instead of backpropagation

- **Representational Power of RBM**
  Any distribution \(\{0,1\}^n\) can be approximated arbitrarily well with an RBM with \(k + 1\) hidden units \([k = \#(\text{input vectors})]\)

\[
P(v, h) = \frac{1}{Z} e^{-E(v, h)}
E(v, h) = -a^T v - b^T h - v^T W h
\]

\[
\text{arg max}_{W} \prod_{v \in V} P(v)
\]
Implementation

- Dependent on Task & Data

classification
- SVC
- Ensemble Classifiers
- KNeighbors Classifier
- SGD Classifier
- Naive Bayes
- Text Data
- Linear SVC

<100K samples

>50 samples

got more data

regression
- SGD Regressor
- Lasso
- ElasticNet

<100K samples

few features should be important

dimensionality reduction
- Randomized PCA
- Isomap
- Spectral Embedding
- LLE

clustering
- Spectral Clustering
- GMM
- KMeans
- MiniBatch KMeans
- MeanShift
- VBGMM

<10K samples

<10K samples

<10K samples

looking

predicting a quantity

do you have labeled data

number of categories known

predicting a category

tough luck

structure

kernel approximation

not working
Implementation

Many Tricks and Details

- Which machine to use?
- How to initialize weights; random or pre-training?
- How many layers, nodes, feature maps?
- How to tune parameters (e.g. learning rate)?
- Which optimization, cost/activation function, pooling?
- How many epochs; how large ratio of mini-batch size?
- How to visualize?
- Which software/hardware?

To prevent overfitting

- Early stopping, cross-validation, regularization (L1/L2, dropout)
Implementation

- Libraries
  - Caffe
  - Torch
  - TensorFlow
  - Theano
LEARNING ABOUT NEURAL NETWORK

How can we interpret what a machine learns?
Explainable AI (XAI)

- Interpretability Problem & Ethics Issue
  - technical challenge of explaining AI decisions

- DARPA project
  - AI whose actions can be easily understood by humans.
  - Contrasted with "black box" AIs

https://www.darpa.mil/program/explainable-artificial-intelligence
Generating Data from NN

Dreaming

- Generative Adversarial Network (GAN)
- Generation of images that produce desired activations in a trained deep network (What does NN really see?)
- Reverse running of a trained NN
  (weights are held fixed and the input is adjusted)
- Google’s DeepDream

The same image before (left) and after (right) applying ten iterations of DeepDream (Wikipedia)
“Torture the data, and it will confess to anything.”
– Ronald Coase
Application to Physics

Based on talks in 2017 Beijing conference
Topics

- Conceptual connections of machine learning and many-body physics
- Machine learning techniques for solving many-body physics/chemistry problems
- Statistical and quantum physics perspectives on machine learning
- Quantum algorithms and quantum hardwares for machine learning
Using ML for Physics Problems

- **ML as a Classifier**
  - A neural network perspective on the Ising gauge theory and the toric code
  - Learning Phase Transitions with/without Confusion
  - Quantum Loop Topography for Machine learning on topological phase, phase transitions, and beyond
  - Machine learning phases of strongly-correlated fermions
  - Magnetic Phase Transitions and Unsupervised Machine Learning
  - Machine Learning for Frustrated Classical Spin Models
  - A Separability-Entanglement Classifier via Machine Learning
  - Transforming Bell's Inequalities into State Classifiers with Machine Learning
Using ML for Physics Problems

- **ML as a Recommender System**
  - Self-Learning Monte Carlo Method
  - Accelerated Monte Carlo simulations with restricted Boltzmann machines
  - Self-Learning quantum Monte Carlo method in interacting fermion systems
  - Self-learning Monte Carlo Method: Continuous Time Algorithm

- **ML for Feature Extraction**
  - Neural-Network Quantum State Tomography for Many-Body Systems
  - Bayesian spectral deconvolution: How many peaks are there in this spectrum?
Using ML for Physics Problems

- ML as a Many-Body Numerical Ansatz
  - Neural-network Quantum States
  - Simulating quantum many-body problems of fermions and quantum spins
  - Efficient Representation of Quantum Many-body States with Deep Neural Networks
  - Machine-learning density functionals
  - Unified Representation for Machine Learning of Molecules and Crystals
  - Machine learning quantum states and entanglement
  - Neural network representation of tensor network and chiral states
  - On the Equivalence of Restricted Boltzmann Machines and Tensor Network States
THANKS!

Any questions?

“People worry that computers will get too smart and take over the world, but the real problem is that they're too stupid and they've already taken over the world.”

— Pedro Domingos