UMD Phys 486 / 786:
Machine learning for physicists

Maissam Barkeshli

UMD

DSECOP Workshop

6/26/2023

Importance of data science education

- There has been immense progress in understanding how to extract knowledge from data over the past century

  Much progress has come from fields of statistics, optimization, information theory, machine learning, …

- These tools have become indispensable in many areas of science and engineering and throughout industry

- Often when physicists go into industry, they work in data science

Physicists are not trained in these subjects in either undergraduate or graduate level through usual curriculum

There is no undergraduate or graduate course that physicists typically take which teaches these ideas

ML / AI research

- Since the 2012 AlexNet moment, ML / AI has become one of the most exciting areas of research
  Many unimaginable breakthroughs over the last decade (computer vision, games, protein folding, nlp)

- There is a growing community of physicists interested in studying science of ML.

  - modern ML systems are complex dynamical systems with 10^6 – 10^{12} learnable parameters.

    Understanding their behavior is a task for physicists

  - Learning and intelligence are some of the most profound natural phenomena

    Should be considered within the purview of physics research

    ML provides toy models to study learning and intelligence.

It is time to start teaching ML seriously as part of the physics curriculum

ASPEN CENTER FOR PHYSICS

May 28 - June 18
Theoretical Physics for Deep Learning

Organizers:
***Maissam Barkeshli,** University of Maryland
Andrey Gromov, Brown University
Alexander Maloney, McGill University
Dan Roberts, Massachusetts Institute of Technology
****Eva Silverstein,** Stanford University
****James Sully,** Anthropic
****Sho Yaida**, Meta AI

The rapid growth in the popularity of deep learning has been fueled by transformational advances in the capabilities of artificial intelligence. There is also considerable interest in applications of machine learning and optimization dynamics to a wide variety of scientific and mathematical problems, including data analysis, novel numerical approaches to partial differential equations, wavefunction approximation in condensed matter and AMO physics, and the discovery and verification of mathematical theorems. We invite applications from scientists interested in research at the intersection of physics and artificial intelligence, and in particular on the construction of physics-inspired models of statistical machine learning, the dynamics of optimization, and the workings of deep neural networks. We welcome applications from across the physics community, including high energy, condensed matter and statistical physicists, as well as computer scientists, machine learning researchers, and mathematicians.

- Phys 486 / 786 was a joint undergraduate and graduate pilot course on machine learning taught in Fall 2023.

  MW 4:00 – 5:15pm
  29 Lectures total (approx. 14 weeks)

  Half TA: Dayal Singh

- Goal was to teach basic concepts and methods in (contemporary) machine learning and make it to modern deep learning

  Blackboard lectures, occasional slides

- Satisfied the undergraduate computational physics requirement

- This course was particularly challenging for the undergraduates (initially conceived of as a grad course)
  Fast-paced concepts + in-depth programming assignments
  Graduate students were comfortable

Enrollment

- Official enrollment (at end of term): 8 undergrads
  22 grads

- Includes registered auditors

  There were also a handful of additional unregistered auditors

  Approx. 25 – 35 students (+1 faculty) attending lecture

- # students who completed coursework for grades (at end of term): 6 undergrads
  12 grads

Course topics

I. Introduction (1 lecture)

II. Classic ML (8 – 10 lectures)

III. Theoretical background: statistics, information theory (5-6 lectures)

IV. Optimization (4-5 lectures)

V. Neural networks (5-6 lectures)

VI. Generative models (2-3 lectures)

Textbooks

Classic ML:

- Hastie, Tibshirani, Friedman, Elements of Statistical Learning Theory
- Gareth, Witten, Hastie, Tibshirani, An introduction to statistical learning with applications in R
- Christopher Bishop, Pattern Recognition and Machine Learning

Neural networks, generative models:

- Goodfellow, Benjio, Courville, Deep Learning, available at https://www.deeplearningbook.org
- Roberts, Yaida, The Principles of Deep Learning Theory, available at https://arxiv.org/abs/2106.10165
- Zhang, Lipton, Li, Smola, Dive into Deep Learning, available at https://arxiv.org/abs/2106.11342
- Murphy, Probabilistic Machine Learning, An introduction, available at https://probml.github.io
- Murphy, Probabilistic Machine Learning, Advanced topics

Optimization:

- Nocedal and Wright, Numerical Optimization

Other material selected from various books, research articles, lecture notes, original derivations

Course topics

I. Introduction

0. Quick survey of some modern ML breakthroughs (AlexNet, AlphaGo, AlphaFold, GPT)

1. Concept of data distribution

   Data drawn i.i.d. from data distribution

   True vs. empirical data distribution

2. Brief overview of basic ideas of supervised, unsupervised, and reinforcement learning

II. Classic machine learning

1. Linear regression

   Mean squared error; L_1 , L_2 regularization; exact solution for ridge regression;

   solving with gradient-based optimization; locally linear regression

2. Linear classifiers: perceptron, logistic regression

3. k-nearest neighbor method for regression and classification. Example of “non-parametric” method

II. Classic machine learning (cont'd)

4. Feature space, kernel trick, kernel regression

Using feature functions to do polynomial regression and nonlinear classification problems

Rewriting linear models in feature space using kernels

Some basics of kernel theory, e.g. reproducing kernel Hilbert spaces

Special selected topic: random Fourier features

5. Support vector machines

Maximizing the margin; hard vs. soft SVM; Lagrange duality and KKT conditions;

sparsity of SVM; multi-class SVM; support vector regression

III. Theoretical background

1. Information theory

   - Entropy
   - Conditional entropy, mutual information
   - Relative entropy (Kullback-Liebler divergence)
   - Fisher information (second derivative of KL divergence)
   - Fisher information as metric on space of probability distributions
   - Other distances between distributions: Jensen-Shannon divergenece, Wasserstein metric and optimal transport

III. Theoretical background (cont'd)

2. Statistics

- Idea of an estimator
- Bias and variance of estimators
- Maximum likelihood estimator
- Origin of loss functions as maximum likelihood estimators (e.g. MSE, cross-entropy)
- Cramer-Rao lower bound (lower bound on variance of estimator in terms of Fisher information)
- Bias-variance decomposition
- Biased estimators (e.g. James-Stein estimator for multi-variate normal distributions)
- Double descent in ML
- Bayesian statistics. Idea of prior, posterior, evidence.
- Maximum a posteriori and relationship between priors and regularization terms
- Model selection, k-fold cross-validation

IV. Optimization

- Gradient descent
- Stochastic gradient descent. Analysis of structure of correlations in SGD noise
- Newton’s method
- Momentum and how it helps reduce dependence of convergence rate on condition number of Hessian
- Nesterov momentum
- Adaptive gradient methods: Adagrad, RMSProp, Adam, AdamW
- Natural gradients (using the Fisher information matrix as a preconditioner)
- Conjugate gradient method
- BFGS (a quasi-Newton method to estimate the Hessian)

V. Neural networks

- Fully connected feedforward neural networks. Backpropagation. Importance of initialization for signal propagation
- Residual connections, normalization (layernorm, batchnorm)
- Convolutional neural networks (FCN + translation invariance and locality. Filters, pooling, padding).
- Recurrent neural networks (seq2seq, vec2seq, seq2vec). LSTM, GRUs, encoder-decoder
- Attention mechanism. Idea of a self-attention layer, scaled dot product attention
- Transformer architecture (encoder only, encoder-decoder, decoder only)
- Summary of results of training large transformer models (e.g. in-context and few-shot learning)

VI. Generative models

- Introduction to why generative models are useful. Idea of modeling the data distribution and sampling from it
- Autoregressive models
- Kernel density estimation
- Diffusion models (2 lectures)

Coursework

- **8 homework assignments**. Assigned roughly once every 1.5-2 weeks

  Assignments consisted of implementing basic ideas from scratch in python (using numpy package) on a mix of synthetic datasets and real datasets

  After implementing methods in numpy, then also used sklearn package to see how to do it easily. For neural networks, we used pytorch

  We used Google Colab, and assignments were submitted electronically as Google Colab files

  Occasionally there were analytical problems as well that were submitted separately

  Graduate students typically given additional analytical homework problems

  TA was instrumental in creating coding assignments that guided students through code, did basic set up of functions, allowing students to fill in blanks.

- **Final project**. Consisted of implementing some ML model on some realistic dataset and writing a 3-5 page report

Programming part of assignments

HW 1. Linear regression: implement via gradient descent and exact solution on a synthetic 1d dataset and also on a real dataset of housing prices; Locally linear regression on synthetic dataset.

HW 2. Linear classification: implement logistic regression for binary classification and multi-class classification on synthetic 2d dataset, flower classification dataset, MNIST, and Ising model dataset. Implement k-nearest neighbor classification on concentric circle synthetic dataset

HW 3. Linear classification in feature space and kernels. Use feature functions to classify concentric circle synthetic dataset. Polynomial regression to fit high order polynomial synthetic dataset. Kernel regression with Gaussian kernel on a synthetic sinusoidal dataset and also using random Fourier features. Kernel regression using Gaussian kernel for weather prediction using dataset of real temperatures

HW 4. SVMs. Hard SVM on synthetic dataset in both primal and dual formulations. Soft SVM on non-linearly separable synthetic dataset. SVM applied to a binary MNIST classification task. Support vector regression on a synthetic nonlinear regression task.

HW 5. Information theory. Extract empirical probability distribution of characters in IMDB movie review dataset. Compute entropy of the distribution. Repeat using a bigram model.

Programming part of assignments (cont'd)

HW 6. Playing with estimators, bias and variance on synthetic datasets. Demonstrating sample-wise double descent in linear regression

HW 7. Neural networks. Implement an FCN 3 ways. (1) in numpy, train on MNIST classification task using SGD on cross-entropy loss. (2) Implement FCN using pytorch autograd (3) Implement using nn.module in pytorch.
Apply it to Ising model classification dataset and CIFAR-10.

Implement CNNs using Pytorch nn.module. Implement on CIFAR-10 and a classification task on galaxy dataset

HW 8. Predicting chaotic Lorenz series using reservoir computers (echo-state networks).
Generating Shakespeare-like text using RNNS and decoder-only Transformers.

Student comments from teaching evaluation

What about the course and/or instruction most enhanced your learning?

Comments
Ahh! I don't know where to begin, this was such an incredible course! (Especially for an undergraduate who is planning to pursue a career centered around machine learning) 1) Thank you (!) for giving the topic a rigorous mathematical treatment, and forcing us to implement these algorithms from scratch from the pure mathematics. It helped me understand it so much better than I would normally. While I did struggle with some of the homeworks, I was happy to struggle as it meant I was learning. Many of the CS courses I'm used to in the material give a brief high level view of the math instead of delving into it. 2) The course had numerous highly detailed homeworks that were direct applications of the course material. Incredibly helpful! 3) The Professor's presentation of the material was organized, neat, efficient, and engaging.
The weekly reviews of the HW by the TA as well as office hours and other meetings. In addition, having notes provided to then reference afterward.
Working out the coding problems and seeing similar examples was very helpful and made it easier to understand how the algorithms and structures worked.
The instructor not only covered the state–of–art machine learning techniques but also the background and theory behind them. I personally like the introduction of statistics and information theory and how they can relate to matching learning. The assignments are very instructive and I feel I gained a lot by implementing the learning algorithm by myself. They demystified the magic word "machine learning" and help me gain confidence in using machine learning as a basic tool in my future data analysis tasks.
Sometime the course offers different viewpoints compared to traditional books/papers in the community.
This course was very hard. It pushed me to improve my coding skills. It was very overwhelming at first, but once I got into the swing of things I was able to keep my head above water. The last two textbooks provided by Zhang and Murphy were helpful as well.
Recommended textbooks and recorded zoom sessions were extremely useful in completing homework; office hours both from the TA and the professor; TA sessions to discuss homework solutions

Concluding thoughts

- Machine learning is an enormous subject.

  Many classic ML topics not covered: Gaussian processes, tree-based methods, random forests, boosting

  Covered almost no unsupervised learning or reinforcement learning, very little in generative models

  Did not have many physics applications

- Ideally there would be 2 courses:

  - A regular undergraduate course focusing mainly on classic ML methods, going more slowly and holding students' hands more, with a much more applied angle

  - A special topics graduate course that covers deep learning; theoretical background in statistics, info theory, optimization; some research-level advances

- Physics department should teach these and not relegate to CS / Engineering departments.