Machine Learning For Physicists…

or ”Facility needs - or chances - seen from the other side”

Arno Candel
CTO H2O.ai
@arnocandel

SLAC ICFA 02/28/18

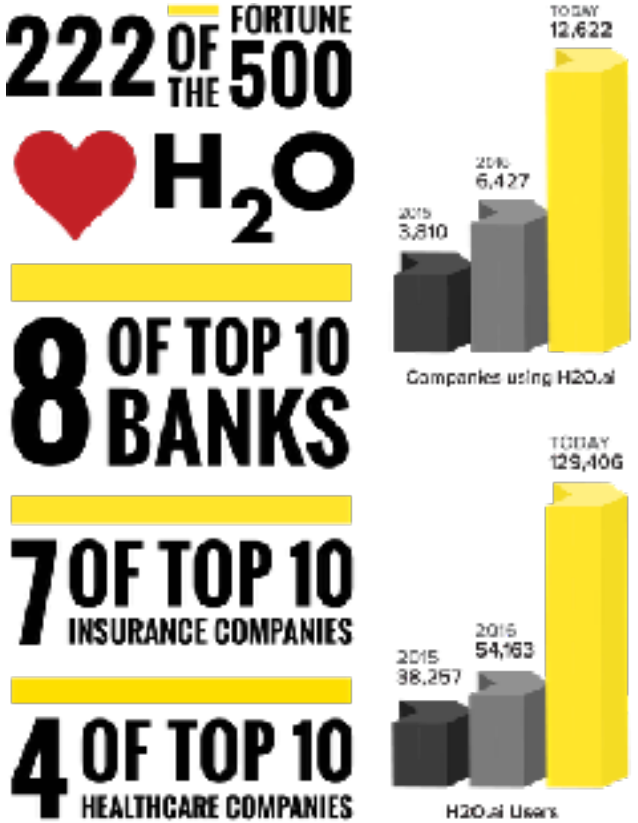
My journey from Physics to Machine Learning

- 2013-now **CTO**, **H2O.ai**, **Machine Learning**, Java/C++/Python/R
- 2012-2013 Senior MTS, Skytree**, Machine Learning**, C++/MPI
- 2005-2012 Staff Scientist, **SLAC** - ACE3P FEM TD & PIC, C++/MPI
- 2005 **PhD Physics** ETH/PSI - Distributed PIC Code, C++/MPI
- 2001 **Masters Physics** ETH, summa cum laude

H2O.ai is a Leader in the Gartner Magic Quadrant

Figure 1. Magic Quadrant for Data Science and Machine-Learning Platforms

Source: Gartner (February 2018)



- Founded in 2011
- 90 employees
- Mountain View, CA
- VC funded (Series C)
- Open-Source Culture

H2O.ai

ML Algorithms are Agnostic

Signal / Noise
Garbage In - Garbage Out
Overfitting / Underfitting
No Free Lunch Theorem

All of this still applies in Physics.
But physicists are creative: opportunities for synergies!

Spring of AI: Machine Learning is Everywhere

nature physics

LETTERS

https://doi.org/10.1038/s41567-018-0048-5

Neural-network quantum state tomography

Giacomo Torlai[1,2], Guglielmo Mazzola[3], Juan Carrasquilla[4,5], Matthias Troyer[3,6], Roger Melko[1,2] and Giuseppe Carleo[3,7]*

my PhD adviser

The experimental realization of increasingly complex synthetic quantum systems calls for the development of general theoretical methods to validate and fully exploit quantum resources. Quantum state tomography (QST) aims to reconstruct the full quantum state from simple measurements, and therefore provides a key tool to obtain reliable analytics[1–3]. However, exact brute-force approaches to QST place a high demand on computational resources, making them unfeasible for anything except small systems[4,5]. Here we show how machine learning techniques can be used to perform QST of highly entangled states with more than a hundred qubits, to a high degree of accuracy. We demonstrate that machine learning allows one to reconstruct traditionally challenging many-body quantities—such as the entanglement entropy—from simple, experimentally accessible measurements. This approach can benefit existing and future generations of devices ranging from quantum computers to ultracold-atom quantum simulators[6–8].

Machine learning methods have been demonstrated to be particularly powerful at compressing high-dimensional data into low-dimensional representations[9,10]. Largely developed in the

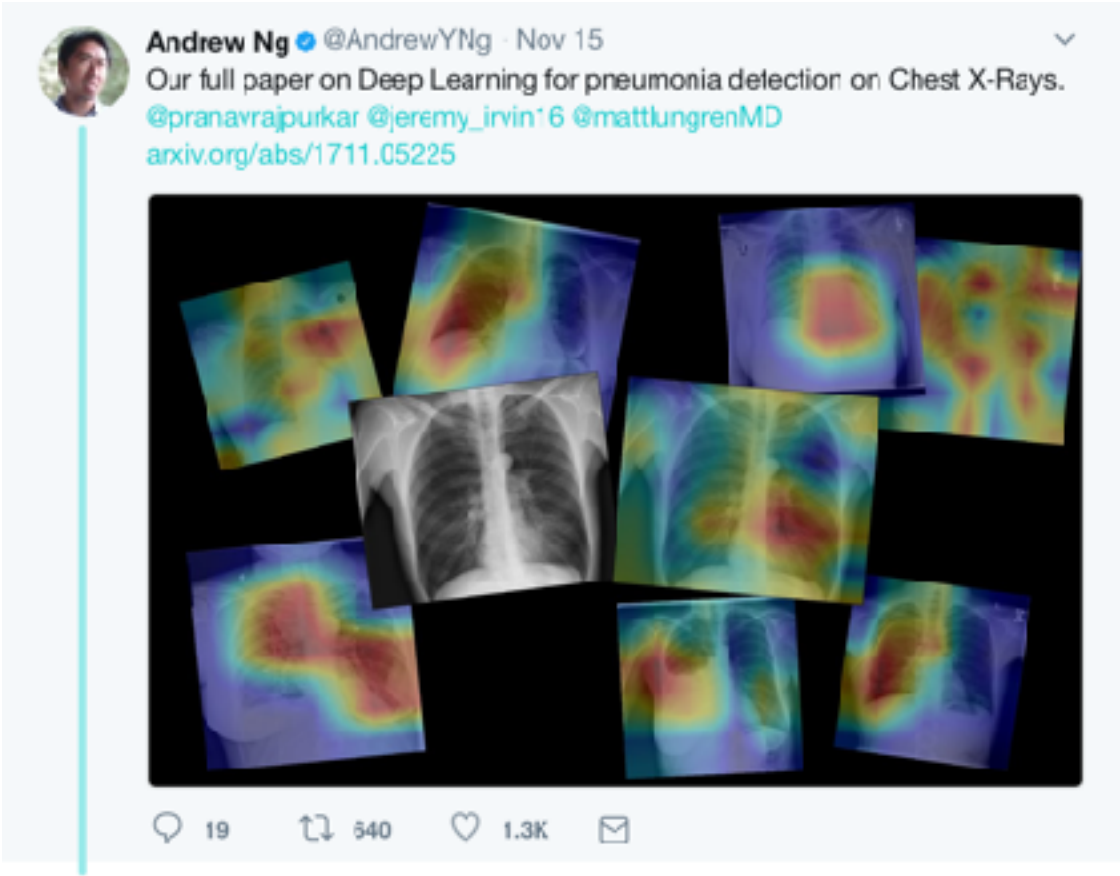
states with a large number of degrees of freedom (qubits, spins and so on), which are thus hard for traditional QST approaches.

We consider here the goal of reconstructing a generic many-body target wavefunction $\Psi(\mathbf{x}) \equiv \langle \mathbf{x}|\Psi\rangle$, where $\mathbf{x}$ is some reference basis (for example, σ^z for spin-1/2). To act as the model, we use a representation of the many-body state in terms of artificial neural networks[11]:

$$\Psi_{\lambda,\mu}(\mathbf{x}) = \sqrt{\frac{p_\lambda(\mathbf{x})}{Z_\lambda}}\, e^{i\phi_\mu(\mathbf{x})/2} \qquad (1)$$

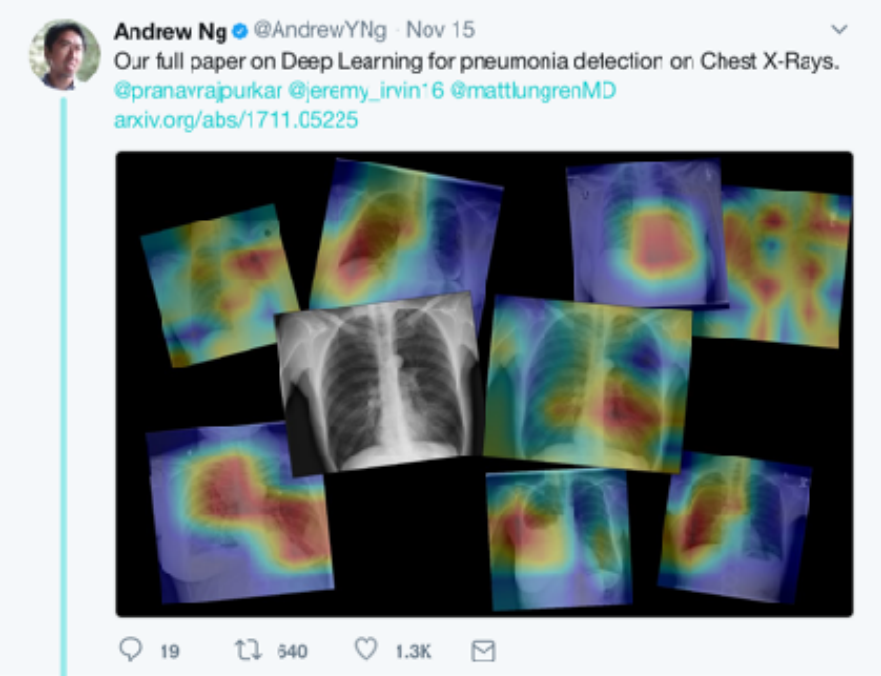
where the networks $p_\lambda(\mathbf{x})$ and $\phi_\mu(\mathbf{x})$ represent, respectively, the amplitude and phase of the state, and Z_λ is the normalization constant. The neural-network architecture we use in this work is based on the restricted Boltzmann machine (RBM). This architecture features a visible layer (describing the physical qubits) and a hidden layer of binary neurons, fully connected with weighted edges to the visible layer (see Methods). RBM states offer a compact variational representation of many-body quantum states, capable of sustaining non-trivial correlations, such as high entanglement,

Spring of AI: Machine Learning is Everywhere



We use the ChestX-ray14 dataset released by Wang et al. (2017) which contains 112,120 frontal-view X-ray images of 30,805 unique patients. Wang et al. (2017) annotate each image with up to 14 different thoracic pathology labels using automatic extraction methods on radiology reports. We label images that have pneumonia as one of the annotated pathologies as positive examples and label all other images as negative examples for the pneumonia detection task. We randomly split the entire dataset into 80% training, and 20% validation.

Looks great, BUT: It's wrong.



Automation needed to avoid human error

Submission history

From: Pranav Rajpurkar [view email]
[v1] Tue, 14 Nov 2017 17:58:50 GMT (16273kb,D)
[v2] Sat, 25 Nov 2017 04:21:27 GMT (321kb,D)

Mistake

We use the ChestX-ray14 dataset released by Wang et al. (2017) which contains 112,120 frontal-view X-ray images of 30,805 unique patients. Wang et al. (2017) annotate each image with up to 14 different thoracic pathology labels using automatic extraction methods on radiology reports. We label images that have pneumonia as one of the annotated pathologies as positive examples and label all other images as negative examples for the pneumonia detection task. We randomly split the entire dataset into 80% training, and 20% validation.

CheXNet (ours)
0.8209
0.9048
0.8831
0.7204
0.8618
0.7766
0.7632
0.8932
0.7939
0.8932
0.9260
0.8044
0.8138
0.9387

CheXNet (ours)
0.8094
0.9248
0.8638
0.7345
0.8676
0.7802
0.7680
0.8887
0.7901
0.8878
0.9371
0.8047
0.8062
0.9164

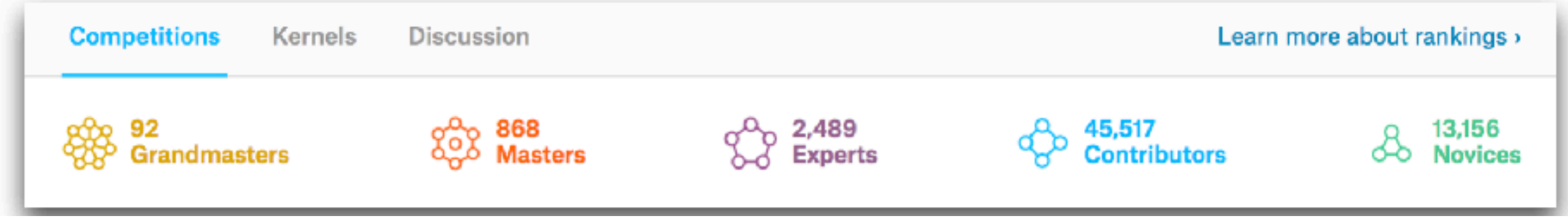
Correction

We use the ChestX-ray14 dataset released by Wang et al. (2017) which contains 112,120 frontal-view X-ray images of 30,805 unique patients. Wang et al. (2017) annotate each image with up to 14 different thoracic pathology labels using automatic extraction methods on radiology reports. We label images that have pneumonia as one of the annotated pathologies as positive examples and label all other images as negative examples. For the pneumonia detection task, we randomly split the dataset into training (28744 patients, 98637 images), validation (1672 patients, 6351 images), and test (389 patients, 420 images). There is no patient overlap between the sets.

Shortage of Data Scientists

"The United States alone faces a shortage of 140,000 to 190,000 people with analytical expertise and 1.5 million managers and analysts"

–McKinsey Prediction for 2018



10,000 hours "coders"

1000 hours "scripters"

100 hours "copy & pasters"

"hello world"

Best way to master Data Science and Machine Learning?

KAGGLE

KAGGLE!

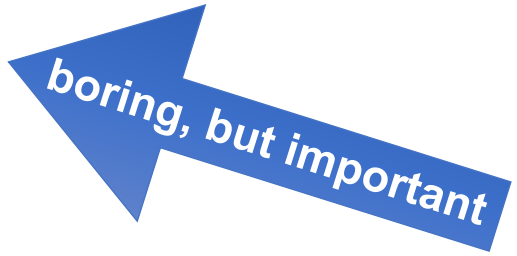
KAGGLE!!

KAGGLE!!!

(seriously, go to http://kaggle.com and spend 100+ hours competing in data science competitions. do it.)

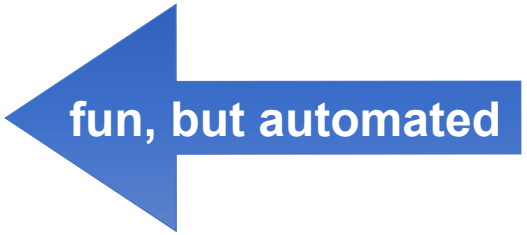
The ~~Key~~ Checklist to Success

- Understand the problem (very very well) - **What is the goal** of applying ML? How is it going to help?
- What is the structure of the data? Tabular? Blobs? Mixed? How **big** is the **data**? 100M rows x 10 cols? 10k rows x 100k cols?
- Do you have labeled data (**supervised)**? Regression or Classification? Why? How can you get more labeled data?
- No labels (**unsupervised)**? Clustering? Autoencoder? Remember not to use Euclidean distance in >> 10 dimensions.
- Find useful metrics (squared error, confusion matrix based, area under ROC curve, etc.)
- Devise a **validation strategy** (how to estimate model's generalization performance on holdout data splits), e.g.:
  75% **train** - model training - historic data to learn from
  15% **valid** - parameter tuning (used repeatedly) - should behave similarly as test data
  10% **test** - final model scoring (used one time only) - simulate production environment
- Do you have access to **GPUs**? 5-50x speedup per GPU (github.com/h2oai/h2o4gpu)
- Pick a platform to train models. R/Py/GUI? Single-node or distributed?
- < 10GB, **structured**: sklearn/H2O-3/XGBoost/LightGBM
- > 10GB, **structured**: Sparkling Water or H2O-3 (github.com/h2oai/h2oai)
- **unstructured** (image/sound/text): **Deep Learning** TensorFlow/PyTorch/Caffe2



iterate until done:

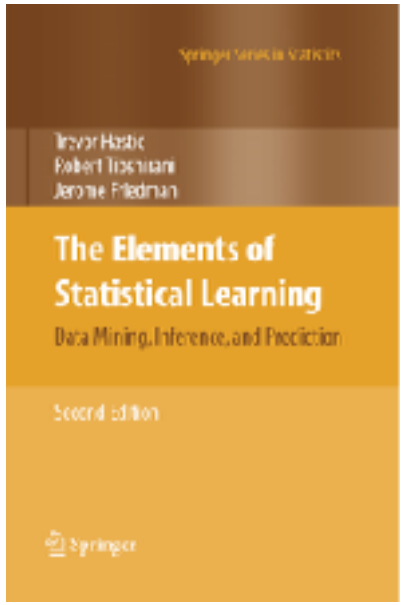
- **engineer features**
- **tune model parameters**
- **ensemble models** (stacking/blending)
- **estimate model performance**



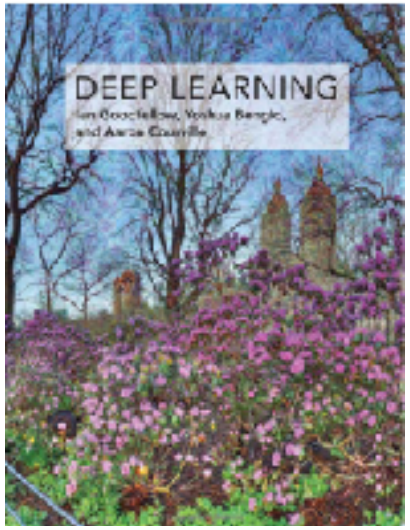
Are you able to explain the problem and the solution to your colleagues and family?

Statistical Learning vs Deep Learning - Need Both!

https://tech.instacart.com/how-to-build-a-deep-learning-model-in-15-minutes-a3684c6f71e



https://web.stanford.edu/~hastie/Papers/ESLII.pdf



http://www.deeplearningbook.org

The “Secret Sauce”: Feature Engineering

From Wikipedia, the free encyclopedia

Feature engineering is the process of using domain knowledge of the data to create features that make machine learning algorithms work. Feature engineering is fundamental to the application of machine learning, and is both difficult and expensive. The need for manual feature engineering can be obviated by automated feature learning.

Feature engineering is an informal topic, but it is considered essential in applied machine learning.

> Coming up with features is difficult, time-consuming, requires expert knowledge. "Applied machine learning" is basically feature engineering.
>
> — Andrew Ng, *Machine Learning and AI via Brain simulations* [1]

H2O.ai Webinar on Feature Engineering

https://www.youtube.com/watch?v=VMTKcT1iHww

H2O Driverless AI: Automated Feature Engineering (AI to do AI)

Kaggle competition from 2016

BNP Paribas Cardif Claims Management

Can you accelerate BNP Paribas Cardif's claims management process?

$30,000 · 2,926 teams · 2 years ago

Submission and Description	Private Score	Public Score
f7d3c656-7711-4cf8-9426-d69a1790c7ec.csv a day ago by Arno Candel 7/10/5 0.43919 +/- 0.0036872	0.43235	0.43475

"2 months for grandmasters — 1 hour for Driverless AI"

In the money · Gold · Silver · Bronze

#	△pub	Team Name	Kernel	Team Members	Score	Entries	Last
1	—	Dexter's Lab			0.42037	198	2y
2	—	escalated chi			0.42079	162	2y
3	—	Exploding Kittens			0.42182	124	2y
4	—	Branden | Nickol | utility			0.42259	251	2y
5	—	the flying burrito brothers			0.42450	264	2y
6	—	n_m			0.42535	4	2y
7	—	PAFY			0.42557	310	2y
8	—	KAME			0.42688	121	2y
9	—	Jack (Japan)			0.42744	22	2y
10	▲1	Dmitry & Bohdan			0.43000	192	2y
11	▲1	Li-Der			0.43005	56	2y
12	▼2	BK3M2PRS			0.43089	338	2y
13	—	x2x4x8			0.43107	55	2y
14	—	Frenchies			0.43145	134	2y
15	▲1	Alns			0.43168	55	2y
16	▼1	maze runners			0.43262	164	2y
17	—	BLR-2			0.43313	129	2y
18	▲3	no one			0.43317	88	2y

Driverless AI: 16th place in private LB (out of 2926)
(top 1% in 1 hour, fully automated, no human input)

Summary

- Use Open-Source ML and Data Science tools such as H2O/TF/sklearn
- Automatic ML can save time and avoid costly mistakes (if done right)
- Use Deep Learning only if needed (images/audio/text) - black box
- Use statistical methods like Boosted Trees otherwise - interpretable
- Use GPUs to accelerate algorithms wherever possible
- See the problem from the algorithm's viewpoint (how to improve?)
- Focus on problem statement (ROI) and model validation/interpretation
- Learn the basics skills and stay competitive at Kaggle (check winning solutions, even several year old competitions are useful)