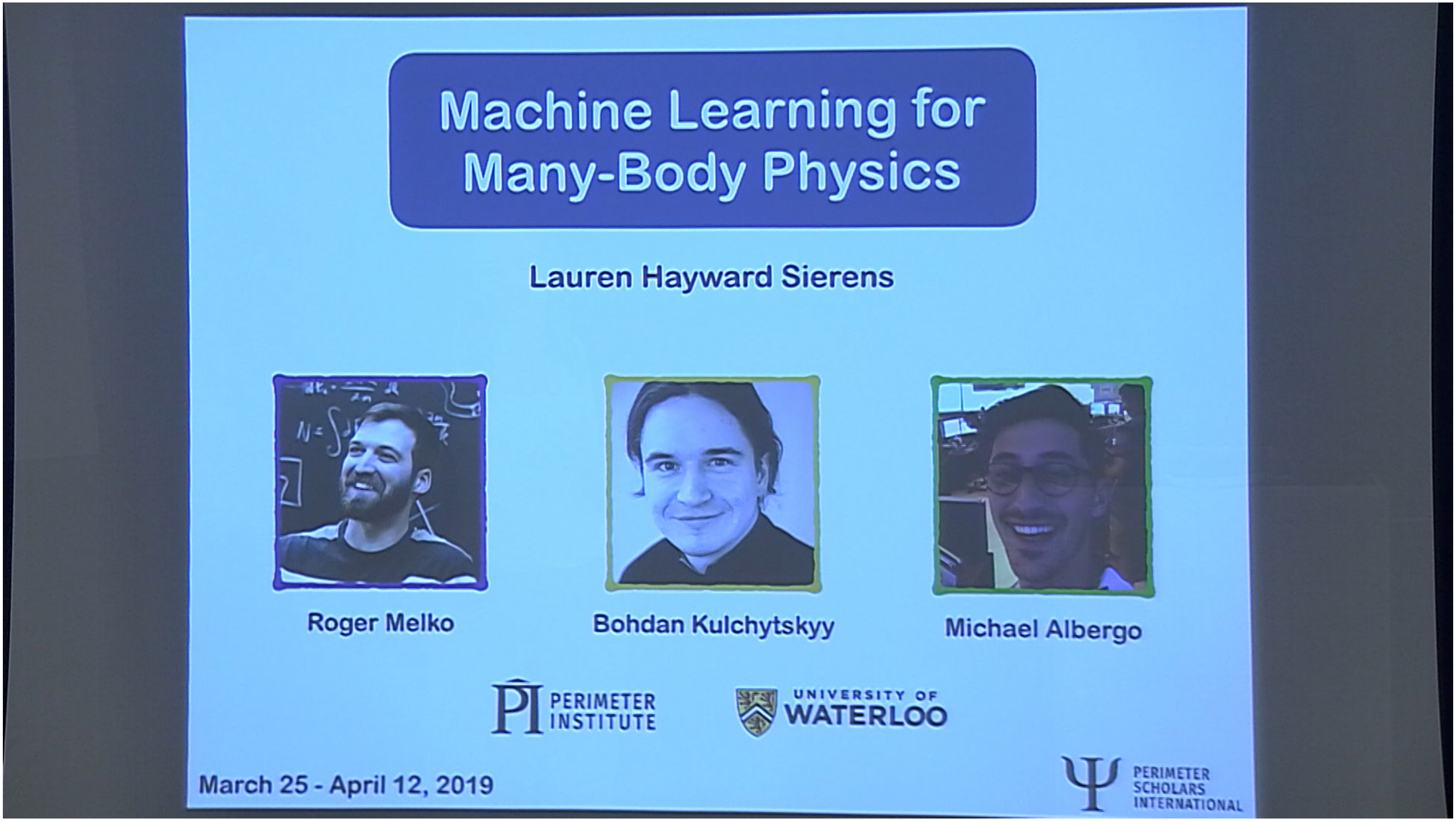
Title: PSI 2018/2019 - Machine Learning - Lecture 1

Speakers: Lauren Hayward Sierens

Collection: PSI 2018/2019 - Machine Learning (Hayward Sierens)

Date: March 25, 2019 - 9:00 AM

URL: http://pirsa.org/19030053


Machine Learning for Many-Body Physics
Lauren Hayward Sierens
Roger Melko
Bohdan Kulchytskyy
Michael Albergo
PERIMETER INSTITUTE
UNIVERSITY OF WATERLOO
March 25 - April 12, 2019
PERIMETER SCHOLARS INTERNATIONAL

Machine learning popularity

PERIMETER SCHOLARS INTERNATIONAL

Image classification

ImageNet Classification with Deep Convolutional Neural Networks

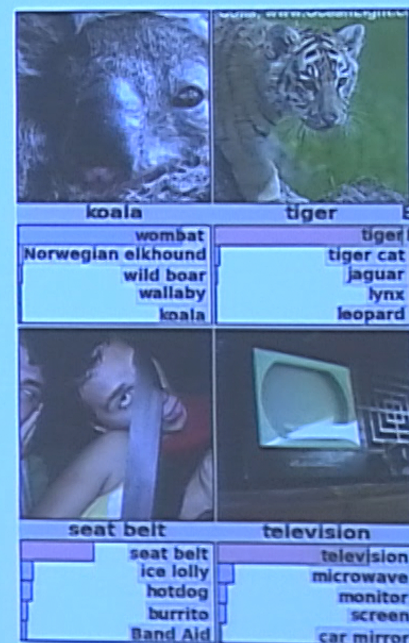
Alex Krizhevsky
University of Toronto
kriz@cs.utoronto.ca

Ilya Sutskever
University of Toronto
ilya@cs.utoronto.ca

Geoffrey E. Hinton
University of Toronto
hinton@cs.utoronto.ca

Abstract

We trained a large, deep convolutional neural network to classify the 1.2 million high-resolution images in the ImageNet LSVRC-2010 contest into the 1000 different classes. On the test data, we achieved top-1 and top-5 error rates of 37.5% and 17.0% which is considerably better than the previous state-of-the-art. The neural network, which has 60 million parameters and 650,000 neurons, consists of five convolutional layers, some of which are followed by max-pooling layers, and three fully-connected layers with a final 1000-way softmax. To make training faster, we used non-saturating neurons and a very efficient GPU implementation of the convolution operation. To reduce overfitting in the fully-connected layers we employed a recently-developed regularization method called "dropout" that proved to be very effective. We also entered a variant of this model in the ILSVRC-2012 competition and achieved a winning top-5 test error rate of 15.3%, compared to 26.2% achieved by the second-best entry.

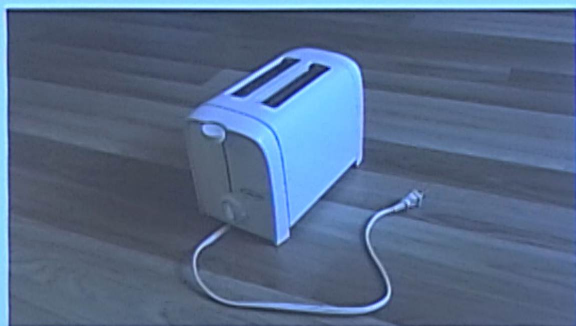


2012

PERIMETER SCHOLARS INTERNATIONAL

Image classification

https://github.com/tensorflow/models



```
toaster (score = 0.99288)
space heater (score = 0.00071)
iPod (score = 0.00034)
printer (score = 0.00024)
pay-phone, pay-station (score = 0.00016)
```

```
mountain bike, all-terrain bike, off-roader (score = 0.80308)
picket fence, paling (score = 0.01216)
bicycle-built-for-two, tandem bicycle, tandem (score = 0.00893)
disk brake, disc brake (score = 0.00346)
sleeping bag (score = 0.00307)
```

PERIMETER SCHOLARS INTERNATIONAL

Image classification

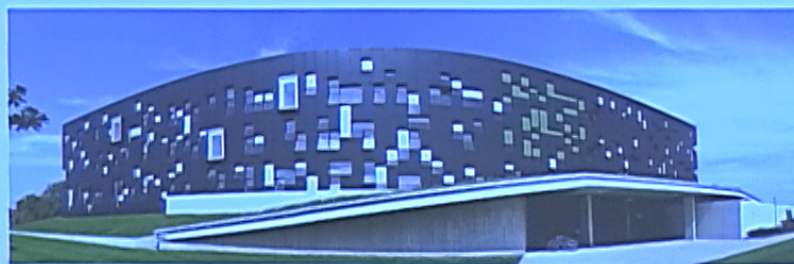
https://github.com/tensorflow/models

```
borzoi, Russian wolfhound (score = 0.46677)
quilt, comforter, comfort, puff (score = 0.06664)
whippet (score = 0.06200)
sleeping bag (score = 0.05488)
Cardigan, Cardigan Welsh corgi (score = 0.02153)
```

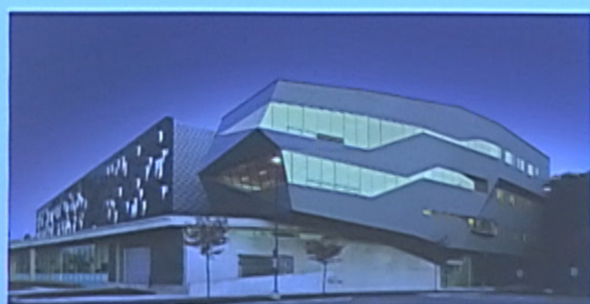
PERIMETER SCHOLARS INTERNATIONAL

Image classification

https://github.com/tensorflow/models



```
barn (score = 0.19133)
planetarium (score = 0.12688)
dome (score = 0.06614)
church, church building (score = 0.05428)
library (score = 0.03213)
```



```
cinema, movie theater, movie theatre, movie house, picture palace (score = 0.47794)
restaurant, eating house, eating place, eatery (score = 0.10927)
planetarium (score = 0.03705)
library (score = 0.03526)
palace (score = 0.03506)
```

Ψ PERIMETER SCHOLARS INTERNATIONAL

AlphaGo

nature

Altmetric: 3193 Citations: 569 More detail »

Article

Mastering the game of Go with deep neural networks and tree search

David Silver, Aja Huang, Chris J. Maddison, Arthur Guez, Laurent Sifre, George van den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Veda Panneershelvam, Marc Lanctot, Sander Dieleman, Dominik Grewe, John Nham, Nal Kalchbrenner, Ilya Sutskever, Timothy Lillicrap, Madeleine Leach, Koray Kavukcuoglu, Thore Graepel & Demis Hassabis

Nature 529, 484–489 (28 January 2016)
doi:10.1038/nature16961
Download Citation

Received: 11 November 2015
Accepted: 05 January 2016
Published: 27 January 2016

Abstract

The game of Go has long been viewed as the most challenging of classic games for artificial intelligence owing to its enormous search space and the difficulty of evaluating board positions and moves. Here we introduce a new approach to computer Go that uses 'value networks' to evaluate board positions and 'policy networks' to select moves. These deep neural networks are trained by a novel combination of supervised learning from human expert games, and reinforcement learning from games of self-play. Without any lookahead search, the neural networks play Go at the level of state-of-the-art Monte Carlo tree search programs that simulate thousands of random games of self-play. We also introduce a new search algorithm that combines Monte Carlo simulation with value and policy networks. Using this search algorithm, our program AlphaGo achieved a 99.8% winning rate against other Go programs, and defeated the human European Go champion by 5 games to 0. This is the first time that a computer program has defeated a human professional player in the full-sized game of Go, a feat previously thought to be at least a decade away.

AlphaGo seals 4-1 victory over Go grandmaster Lee Sedol

DeepMind's artificial intelligence astonishes fans to defeat human opponent and offers evidence computer software has mastered a major challenge

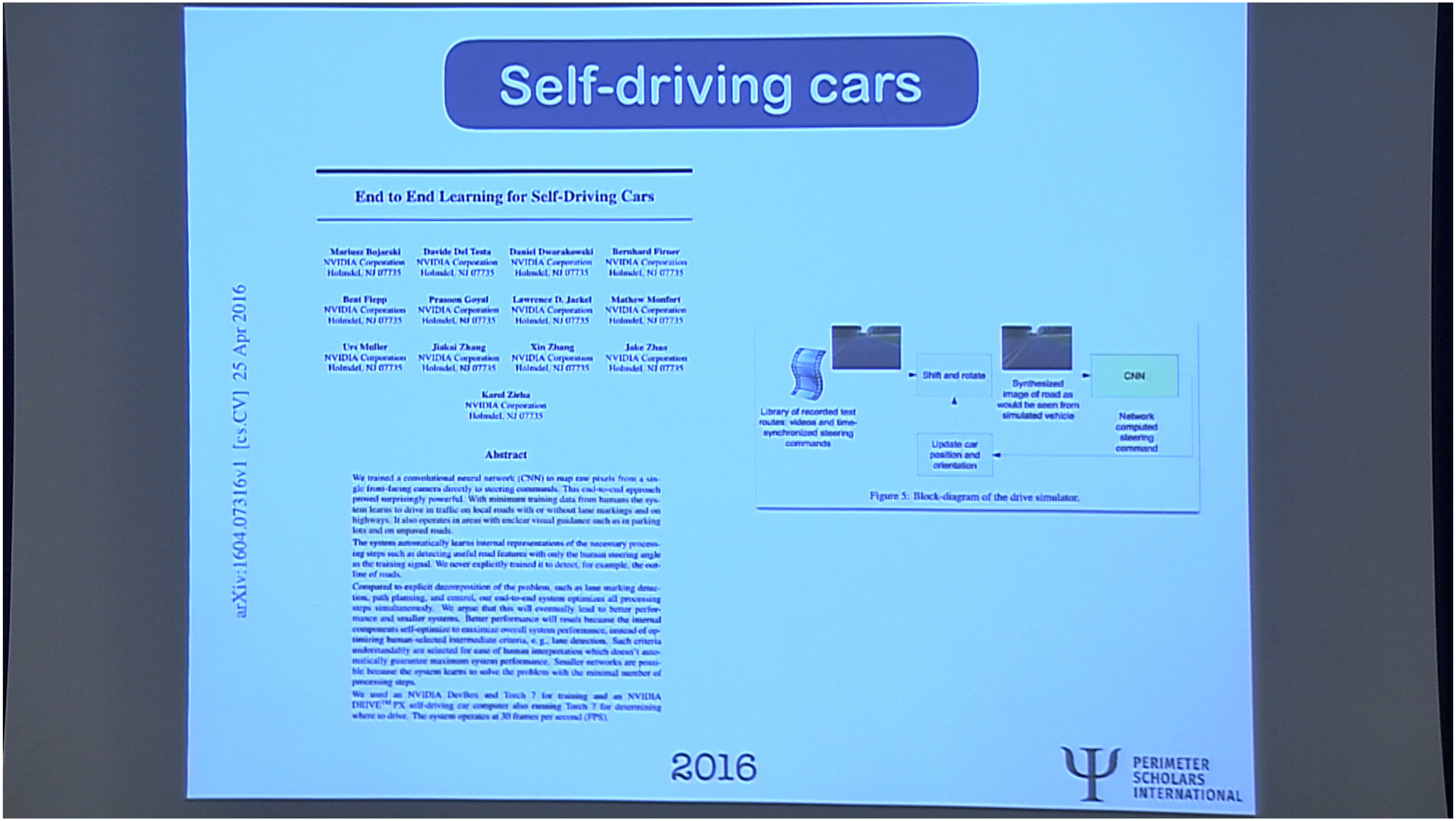
Steven Borowiec

Tue 15 Mar 2016 10.12 GMT

https://www.theguardian.com

2016

PERIMETER SCHOLARS INTERNATIONAL



Self-driving cars
arXiv:1604.07316v1 [cs.CV] 25 Apr 2016
End to End Learning for Self-Driving Cars
Mariusz Bojarski NVIDIA Corporation Holmdel, NJ 07735
Davide Del Testa NVIDIA Corporation Holmdel, NJ 07735
Daniel Dworakowski NVIDIA Corporation Holmdel, NJ 07735
Bernhard Firner NVIDIA Corporation Holmdel, NJ 07735
Beat Flepp NVIDIA Corporation Holmdel, NJ 07735
Prasoon Goyal NVIDIA Corporation Holmdel, NJ 07735
Lawrence D. Jackel NVIDIA Corporation Holmdel, NJ 07735
Mathew Monfort NVIDIA Corporation Holmdel, NJ 07735
Urs Muller NVIDIA Corporation Holmdel, NJ 07735
Jiakai Zhang NVIDIA Corporation Holmdel, NJ 07735
Xin Zhang NVIDIA Corporation Holmdel, NJ 07735
Jake Zhao NVIDIA Corporation Holmdel, NJ 07735
Karol Zieba NVIDIA Corporation Holmdel, NJ 07735
Abstract
We trained a convolutional neural network (CNN) to map raw pixels from a single front-facing camera directly to steering commands. This end-to-end approach proved surprisingly powerful. With minimum training data from humans the system learns to drive in traffic on local roads with or without lane markings and on highways. It also operates in areas with unclear visual guidance such as in parking lots and on unpaved roads.
The system automatically learns internal representations of the necessary processing steps such as detecting useful road features with only the human steering angle as the training signal. We never explicitly trained it to detect, for example, the outline of roads.
Compared to explicit decomposition of the problem, such as lane marking detection, path planning, and control, our end-to-end system optimizes all processing steps simultaneously. We argue that this will eventually lead to better performance and smaller systems. Better performance will result because the internal components self-optimize to maximize overall system performance, instead of optimizing human-selected intermediate criteria, e. g., lane detection. Such criteria understandably are selected for ease of human interpretation which doesn't automatically guarantee maximum system performance. Smaller networks are possible because the system learns to solve the problem with the minimal number of processing steps.
We used an NVIDIA DevBox and Torch 7 for training and an NVIDIA DRIVE™ PX self-driving car computer also running Torch 7 for determining where to drive. The system operates at 30 frames per second (FPS).
Library of recorded test routes: videos and time-synchronized steering commands
Shift and rotate
Synthesized image of road as would be seen from simulated vehicle
CNN
Network computed steering command
Update car position and orientation
Figure 5: Block-diagram of the drive simulator.
2016
PERIMETER SCHOLARS INTERNATIONAL

Language translation

Google's Neural Machine Translation System: Bridging the Gap between Human and Machine Translation

Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V. Le, Mohammad Norouzi
yonghui,schuster,zhifengc,qvl,mnorouzi@google.com

Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, Jeff Klingner, Apurva Shah, Melvin Johnson, Xiaobing Liu, Łukasz Kaiser, Stephan Gouws, Yoshikiyo Kato, Taku Kudo, Hideto Kazawa, Keith Stevens, George Kurian, Nishant Patil, Wei Wang, Cliff Young, Jason Smith, Jason Riesa, Alex Rudnick, Oriol Vinyals, Greg Corrado, Macduff Hughes, Jeffrey Dean

arXiv:1609.08144v2 [cs.CL] 8 Oct 2016

Abstract

Neural Machine Translation (NMT) is an end-to-end learning approach for automated translation, with the potential to overcome many of the weaknesses of conventional phrase-based translation systems. Unfortunately, NMT systems are known to be computationally expensive both in training and in translation inference – sometimes prohibitively so in the case of very large data sets and large models. Several authors have also charged that NMT systems lack robustness, particularly when input sentences contain rare words. These issues have hindered NMT's use in practical deployments and services, where both accuracy and speed are essential. In this work, we present GNMT, Google's Neural Machine Translation system, which attempts to address many of these issues. Our model consists of a deep LSTM network with 8 encoder and 8 decoder layers using residual connections as well as attention connections from the decoder network to the encoder. To improve parallelism and therefore decrease training time, our attention mechanism connects the bottom layer of the decoder to the top layer of the encoder. To accelerate the final translation speed, we employ low-precision arithmetic during inference computations. To improve handling of rare words, we divide words into a limited set of common sub-word units ("wordpieces") for both input and output. This method provides a good balance between the flexibility of "character"-delimited models and the efficiency of "word"-delimited models, naturally handles translation of rare words, and ultimately improves the overall accuracy of the system. Our beam search technique employs a length-normalization procedure and uses a coverage penalty, which encourages generation of an output sentence that is most likely to cover all the words in the source sentence. To directly optimize the translation BLEU scores, we consider refining the models by using reinforcement learning, but we found that the improvement in the BLEU scores did not reflect in the human evaluation. On the WMT'14 English-to-French and English-to-German benchmarks, GNMT achieves competitive results to state-of-the-art. Using a human side-by-side evaluation on a set of isolated simple sentences, it reduces translation errors by an average of 60% compared to Google's phrase-based production system.

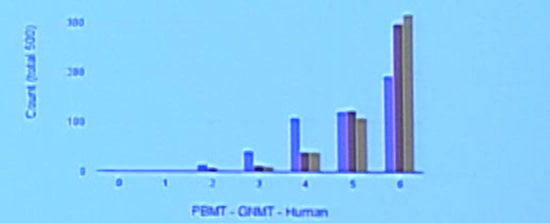
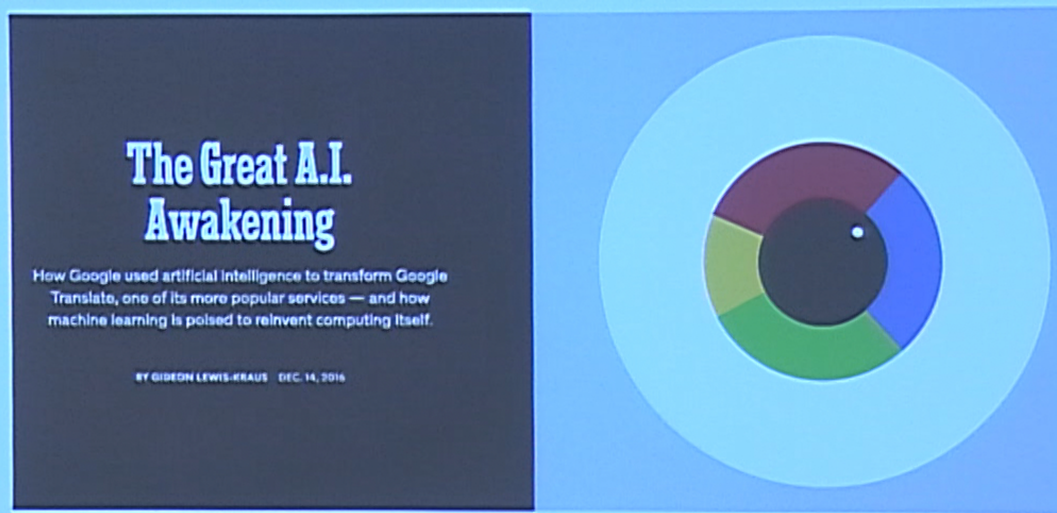


Figure 6: Histogram of side-by-side scores on 500 sampled sentences from Wikipedia and news websites for a typical language pair, here English → Spanish (PBMT blue, GNMT red, Human orange). It can be seen that there is a wide distribution in scores, even for the human translation when rated by other humans, which shows how ambiguous the task is. It is clear that GNMT is much more accurate than PBMT.

2016

PERIMETER SCHOLARS INTERNATIONAL

The New York Times



Translation #1:

Kilimanjaro is 19,710 feet of the mountain covered with snow, and it is said that the highest mountain in Africa. Top of the west, "Ngaje Ngai" in the Maasai language, has been referred to as the house of God. The top close to the west, there is a dry, frozen carcass of a leopard. Whether the leopard had what the demand at that altitude, there is no that nobody explained.

Translation #2:

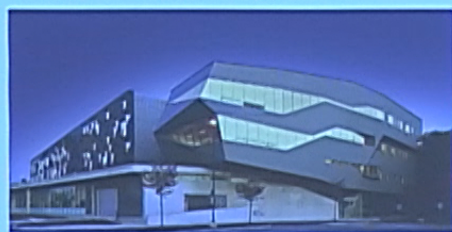
Kilimanjaro is a mountain of 19,710 feet covered with snow and is said to be the highest mountain in Africa. The summit of the west is called "Ngaje Ngai" in Masai, the house of God. Near the top of the west there is a dry and frozen dead body of leopard. No one has ever explained what leopard wanted at that altitude.

PERIMETER SCHOLARS INTERNATIONAL

Generating art

https://deepart.io/

Step 1: Upload photo



Step 2: Choose style

PERIMETER
SCHOLARS
INTERNATIONAL

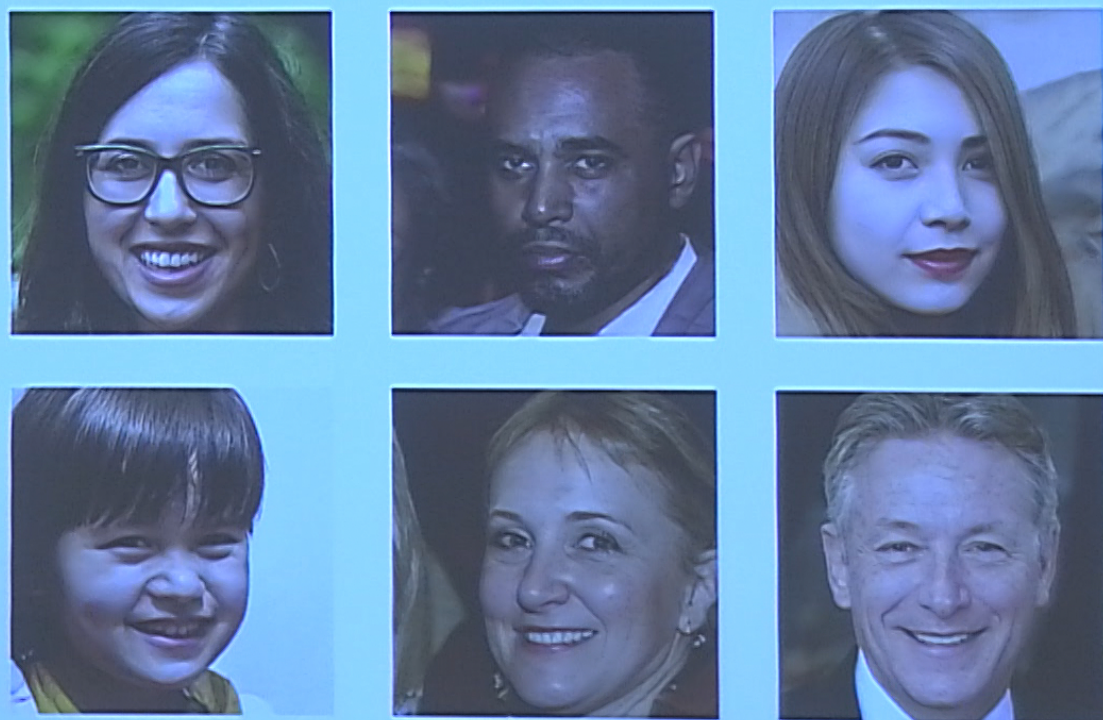
Generating art

Robbie Barrat
https://robbiebarrat.github.io

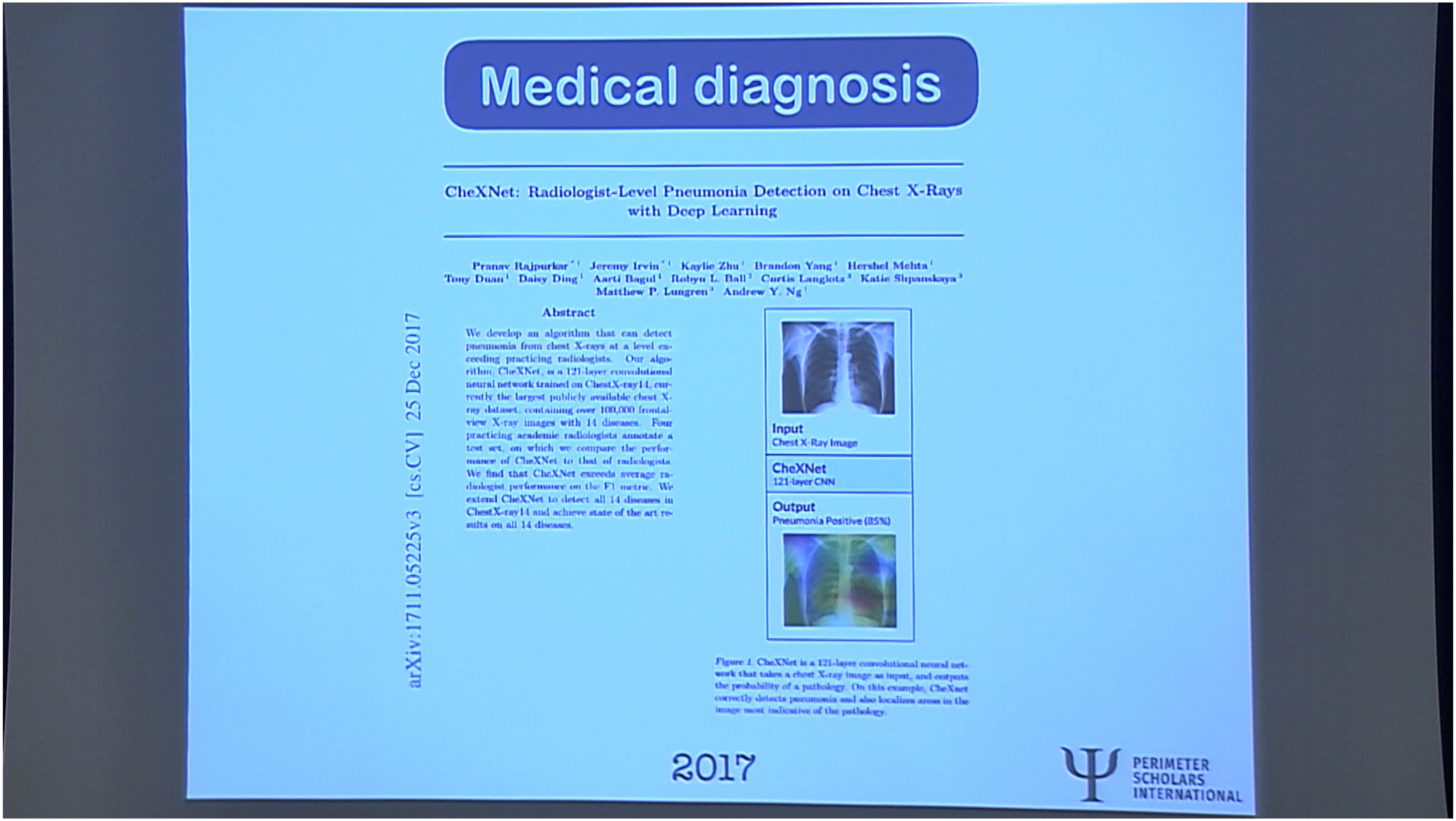
PERIMETER SCHOLARS INTERNATIONAL

Computer-generated people

https://thispersondoesnotexist.com



PERIMETER SCHOLARS INTERNATIONAL


Medical diagnosis
CheXNet: Radiologist-Level Pneumonia Detection on Chest X-Rays with Deep Learning
Pranav Rajpurkar *1 Jeremy Irvin *1 Kaylie Zhu 1 Brandon Yang 1 Hershel Mehta 1
Tony Duan 1 Daisy Ding 1 Aarti Bagul 1 Robyn L. Ball 2 Curtis Langlotz 3 Katie Shpanskaya 3
Matthew P. Lungren 3 Andrew Y. Ng 1
arXiv:1711.05225v3 [cs.CV] 25 Dec 2017
Abstract
We develop an algorithm that can detect pneumonia from chest X-rays at a level exceeding practicing radiologists. Our algorithm, CheXNet, is a 121-layer convolutional neural network trained on ChestX-ray14, currently the largest publicly available chest X-ray dataset, containing over 100,000 frontal-view X-ray images with 14 diseases. Four practicing academic radiologists annotate a test set, on which we compare the performance of CheXNet to that of radiologists. We find that CheXNet exceeds average radiologist performance on the F1 metric. We extend CheXNet to detect all 14 diseases in ChestX-ray14 and achieve state of the art results on all 14 diseases.
Input
Chest X-Ray Image
CheXNet
121-layer CNN
Output
Pneumonia Positive (85%)
Figure 1. CheXNet is a 121-layer convolutional neural network that takes a chest X-ray image as input, and outputs the probability of a pathology. On this example, CheXnet correctly detects pneumonia and also localizes areas in the image most indicative of the pathology.
2017
PERIMETER
SCHOLARS
INTERNATIONAL

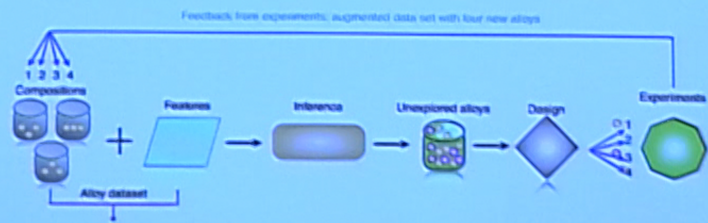
Machine Learning for Physics

nature COMMUNICATIONS

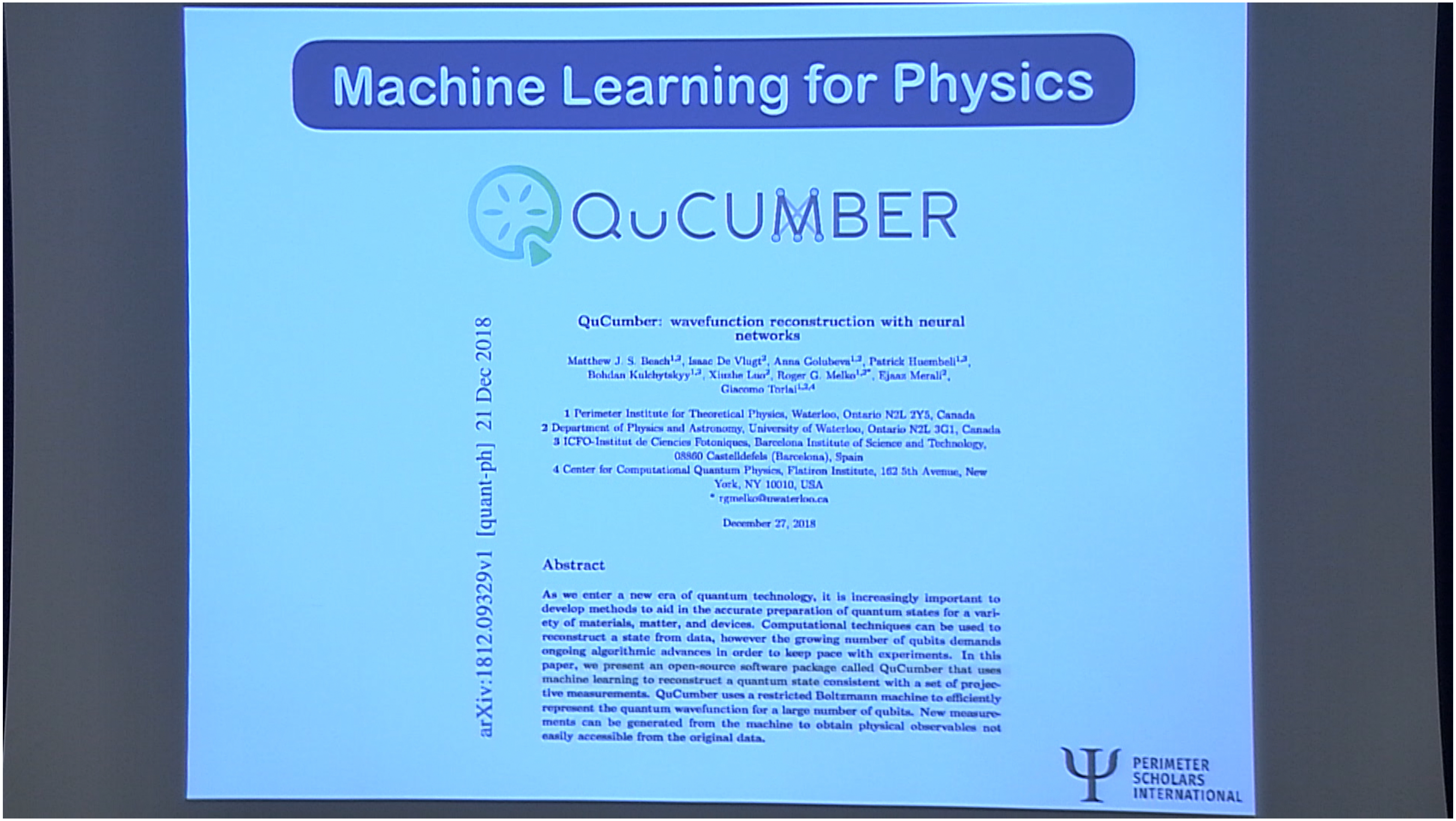
Accelerated search for materials with targeted properties by adaptive design

Dezhen Xue[1,2], Prasanna V. Balachandran[1], John Hogden[3], James Theiler[4], Deqing Xue[2] & Turab Lookman[1]

Finding new materials with targeted properties has traditionally been guided by intuition, and trial and error. With increasing chemical complexity, the combinatorial possibilities are too large for an Edisonian approach to be practical. Here we show how an adaptive design strategy, tightly coupled with experiments, can accelerate the discovery process by sequentially identifying the next experiments or calculations, to effectively navigate the complex search space. Our strategy uses inference and global optimization to balance the trade-off between exploitation and exploration of the search space. We demonstrate this by finding very low thermal hysteresis (ΔT) NiTi-based shape memory alloys, with $Ti_{50.0}Ni_{46.7}Cu_{0.8}Fe_{2.3}Pd_{0.2}$ possessing the smallest ΔT (1.84 K). We synthesize and characterize 36 predicted compositions (9 feedback loops) from a potential space of ~800,000 compositions. Of these, 14 had smaller ΔT than any of the 22 in the original data set.



PERIMETER SCHOLARS INTERNATIONAL


Machine Learning for Physics
QuCUMBER
arXiv:1812.09329v1 [quant-ph] 21 Dec 2018
QuCumber: wavefunction reconstruction with neural networks
Matthew J. S. Beach[1,2], Isaac De Vlugt[2], Anna Golubeva[1,2], Patrick Huembeli[1,3], Bohdan Kulchytskyy[1,2], Xiuzhe Luo[2], Roger G. Melko[1,2*], Ejaaz Merali[2], Giacomo Torlai[1,2,4]
1 Perimeter Institute for Theoretical Physics, Waterloo, Ontario N2L 2Y5, Canada
2 Department of Physics and Astronomy, University of Waterloo, Ontario N2L 3G1, Canada
3 ICFO-Institut de Ciencies Fotoniques, Barcelona Institute of Science and Technology, 08860 Castelldefels (Barcelona), Spain
4 Center for Computational Quantum Physics, Flatiron Institute, 162 5th Avenue, New York, NY 10010, USA
* rgmelko@uwaterloo.ca
December 27, 2018
Abstract
As we enter a new era of quantum technology, it is increasingly important to develop methods to aid in the accurate preparation of quantum states for a variety of materials, matter, and devices. Computational techniques can be used to reconstruct a state from data, however the growing number of qubits demands ongoing algorithmic advances in order to keep pace with experiments. In this paper, we present an open-source software package called QuCumber that uses machine learning to reconstruct a quantum state consistent with a set of projective measurements. QuCumber uses a restricted Boltzmann machine to efficiently represent the quantum wavefunction for a large number of qubits. New measurements can be generated from the machine to obtain physical observables not easily accessible from the original data.
PERIMETER SCHOLARS INTERNATIONAL

What is machine learning?

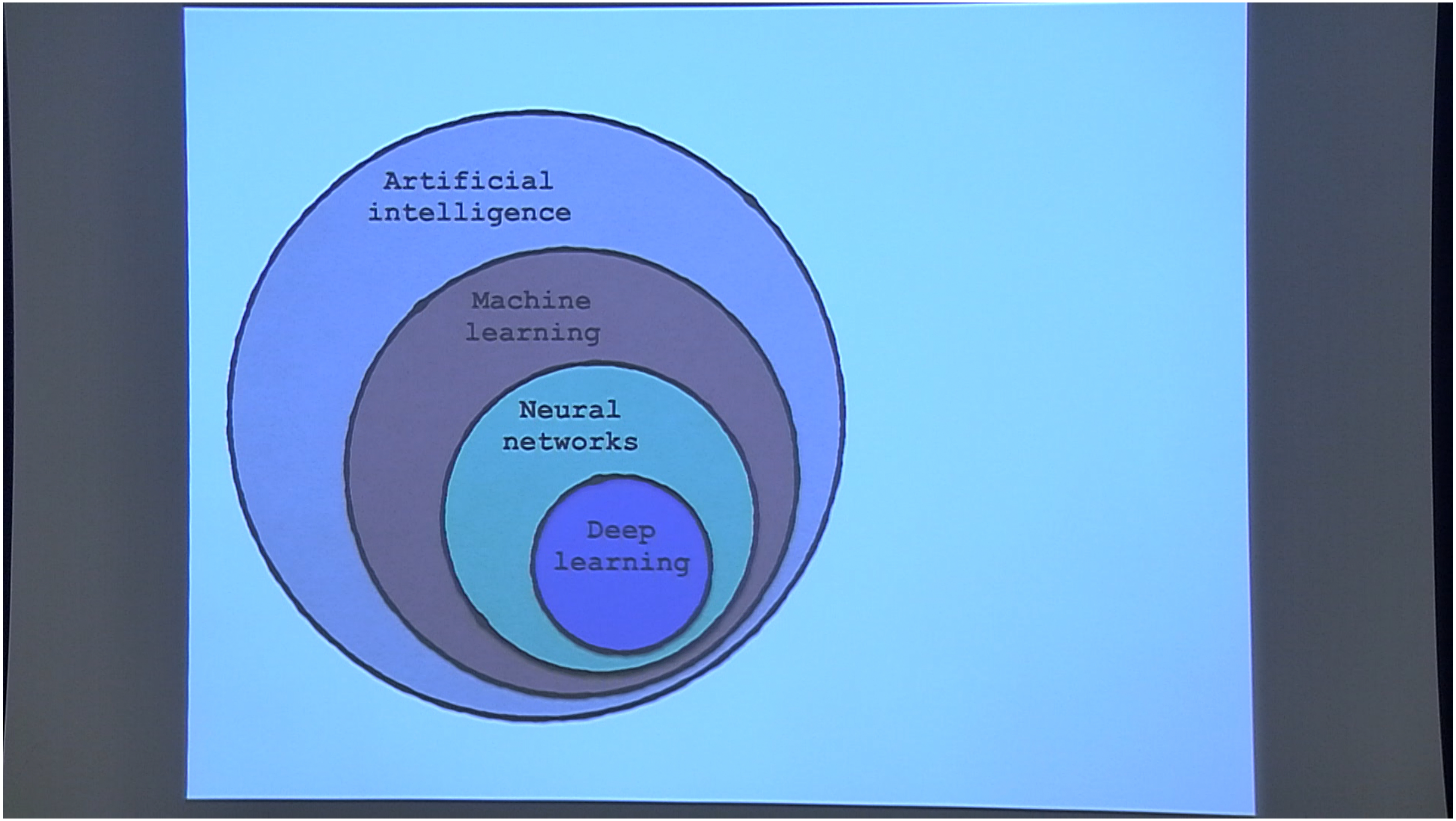
"Machine learning is a field of computer science that uses statistical techniques to give computer systems the ability to "learn" (i.e., progressively improve performance on a specific task) with data, without being explicitly programmed."

https://en.wikipedia.org

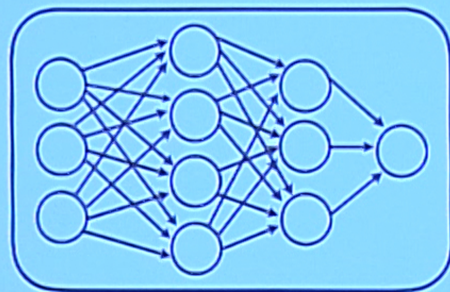
"[Machine learning] is about finding out regularities in data and making use of them for fun and profit."

L.-G. Liu, S.-H. Li and L. Wang, http://wangleiphy.github.io

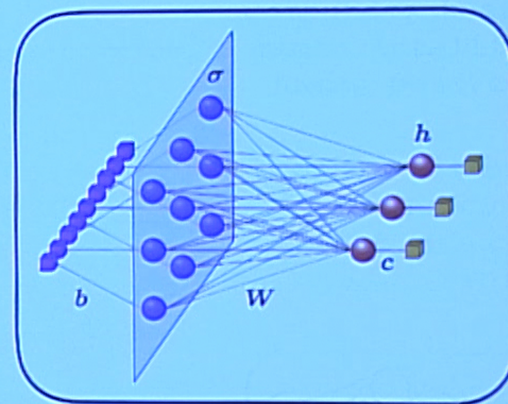
PERIMETER SCHOLARS INTERNATIONAL


Artificial
intelligence
Machine
learning
Neural
networks
Deep
learning

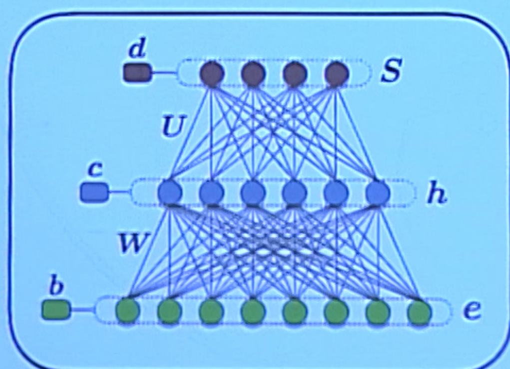
Artificial neural networks



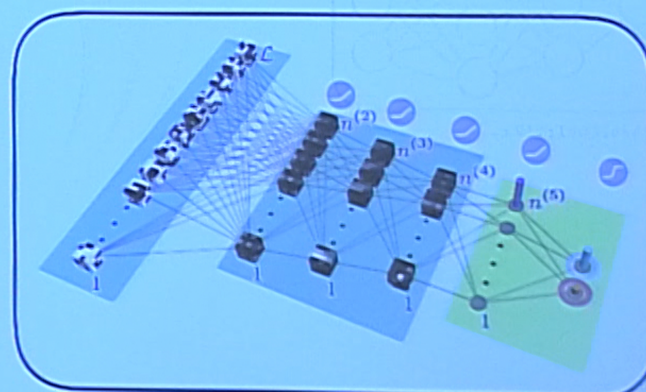
arXiv:1803.08823



arXiv:1606.02718



arXiv:1610.04238



arXiv:1609.02552

PERIMETER SCHOLARS INTERNATIONAL

Machine Learning for Many-Body Physics
Spring 2019
Lauren Hayward Sierens

Homework 1:
Blog post and presentation on
$\langle$ PHYSICS | MACHINE LEARNING $\rangle$

Due dates:
Friday, April 5, 2019 (topic choice)
Friday, April 12, 2019 (blog post and presentation)

The objective of this homework assignment is to write a blog-style article and give a five-minute presentation about current ideas and research within the intersection of physics and machine learning.

You can choose to write and present about recent work done by other researchers, or you can come up with your own ideas for interdisciplinary research combining physics and machine learning (see below for more information and suggestions). For those students who are writing their PSI essay on topics related to machine learning, you must choose a different topic for this assignment. You must inform Lauren of your topic by Friday, April 5, and each student must choose a different topic. The target audience for your article and presentation should be your classmates within this course.

Blog post:
Your article should be at least 1000 words in length, and you are encouraged to incorporate creative visualizations and animations. A PDF copy of the article as well as all documents required to generate it must be submitted in a single ZIP file by Friday, April 12. For inspiration, you may wish to read through posts on blogs such as

- `https://physicsml.github.io/category/articles.html`,
- `https://www.ethz.ch/content/specialinterest/phys/theoretical-physics/cmta/en.html`,
- `http://physhik.com/posts`,
- `https://calculatedcontent.com` (some posts are physics related),
- `https://deepmind.com/blog` (not physics related),
- `https://distill.pub` (not physics related).

Presentation:
On Friday, April 12, we will meet at 1:30pm in the Time Room for the final tutorial of this course. During this tutorial, you will give a five-minute presentation to summarize your blog post and then answer questions from your classmates. You may choose to use slides and/or the blackboard for your presentation.

Topic option 1: Summarize a recent research paper or idea

As a first option, you may choose to write about recent work done by other researchers in article(s) and/or seminar presentation(s). To come up with ideas, you may wish to look through resources such as

- `https://physicsml.github.io/pages/papers.html` (list of relevant papers),
- `http://pirsa.org/3029` (talks within the "Machine Learning Initiative" at PI),
- `http://online.kitp.ucsb.edu/online/machine19` (talks from the "Machine Learning for Quantum Many-Body Physics" program that recently took place at KITP in Santa Barbara),
- `http://online.kitp.ucsb.edu/online/machine-c19` (talks from the conference "At the Crossroad of Physics and Machine Learning" that recently took place at KITP in Santa Barbara),
- `http://wangleiphy.github.io/talks.html` (look at talks from 2017 onwards),

or you can discuss other options with Lauren.

Note that some of the talks included in the links above (including many of those from the KITP program labelled as "Tutorial") address topics in either machine learning or physics but have $\langle$ PHYSICS | MACHINE LEARNING $\rangle \approx 0$. You must be sure to choose a topic for which this overlap is nonzero.

Topic option 2: Write about your own ideas

As another option, you can get creative and write about your own ideas or work. Much of the research within the intersection of physics and machine learning can be divided into two categories. The first category uses existing techniques from machine learning to study problems in physics (such as phase transitions in many-body physics). Perhaps you have ideas for some areas of physics research that could benefit from using machine learning as a tool. The second category of research uses ideas from theoretical physics (such as tensor networks and renormalization group theory) to improve machine learning methods and algorithms. You may develop ideas for how to use intuition and theory from physics to explain phenomena in machine learning.

LHayward@perimeterinstitute.ca

- Machine Learning (ML): training computers to detect and characterize features from data

- Many-body physics: predicting and explaining $\underbrace{\text{macroscopic phenomena}}_{\text{features}}$ from $\underbrace{\text{microscopic quantities}}_{\text{data}}$

Points about many-body physics

↳ fundamentally rely on many particles interacting

↳ needed to explain high-temp SC, fractional quantum Hall effect, features of many qubit quantum computers

↳ not needed to describe electron conduction, semiconductors, etc.
(where effective single-quasiparticle theories exist)

Computational methods are useful for MB physics because solving the Schrödinger eq. is exponentially difficult, quickly becomes intractable as the # of particles increases.

Examples of methods:

↳ Monte Carlo simulation
↳ tensor networks
↳ exact diagonalization

} start from a microscopic theory, then generate data describing the state

ML: more fundamentally data driven

We will study three categories of ML algorithms:

1. Supervised Learning (SL): LECTURES #1-6

Dataset $\mathcal{D} = \{(\vec{x}, \vec{y})\}$

↳ datapoints $\vec{x} = (x_1, x_2, \ldots, x_d)$

↳ labels $\vec{y} = (y_1, y_2, \ldots, y_{d_y})$

driven

...es of ML algorithms:

...L): LECTURES #1–6

Task: fit some function $\vec{f}(\vec{x})$ to $\vec{y}$

↳ "regression" when labels are continuous

↳ "classification" when labels are discrete

$, x_2, \ldots, x_{d_x})$

$y_2, \ldots, y_{d_y})$

(2) Unsupervised Learning (UL): LECTURES 7, 8, $\underbrace{11\text{–}15}_{\text{generative modelling}}$

Dataset $\mathcal{D} = \{\vec{x}\}$ (unlabelled data)

Task: extract meaningful features from the data to efficiently represent the prob. dist. $p_{data}(\vec{x})$

③ Reinforcement Learning (RL) : LECTURES #9, 10

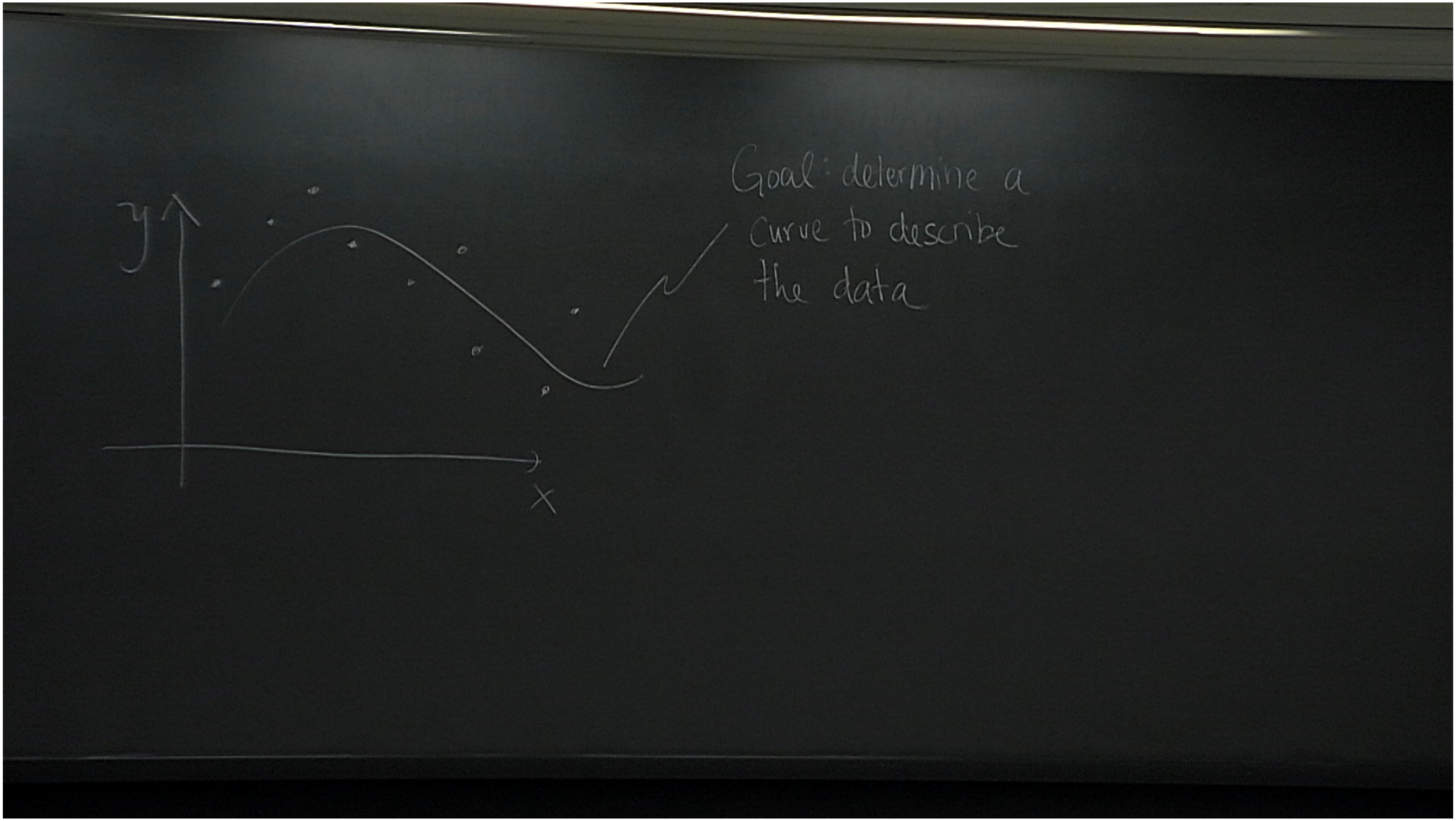
Task: given an environment (state), take an action such that the "reward" is maximized.

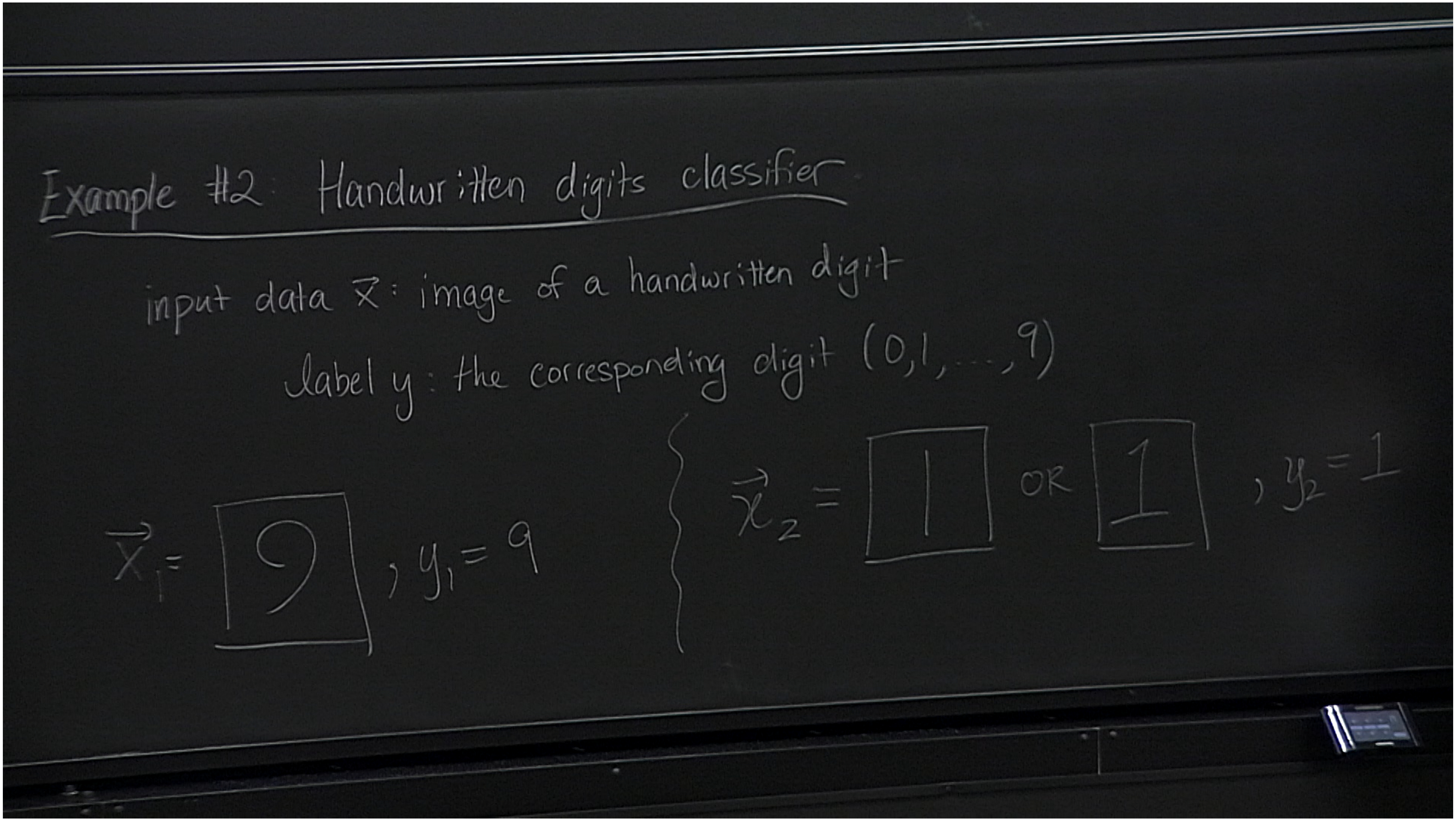
Supervised Learning Examples:

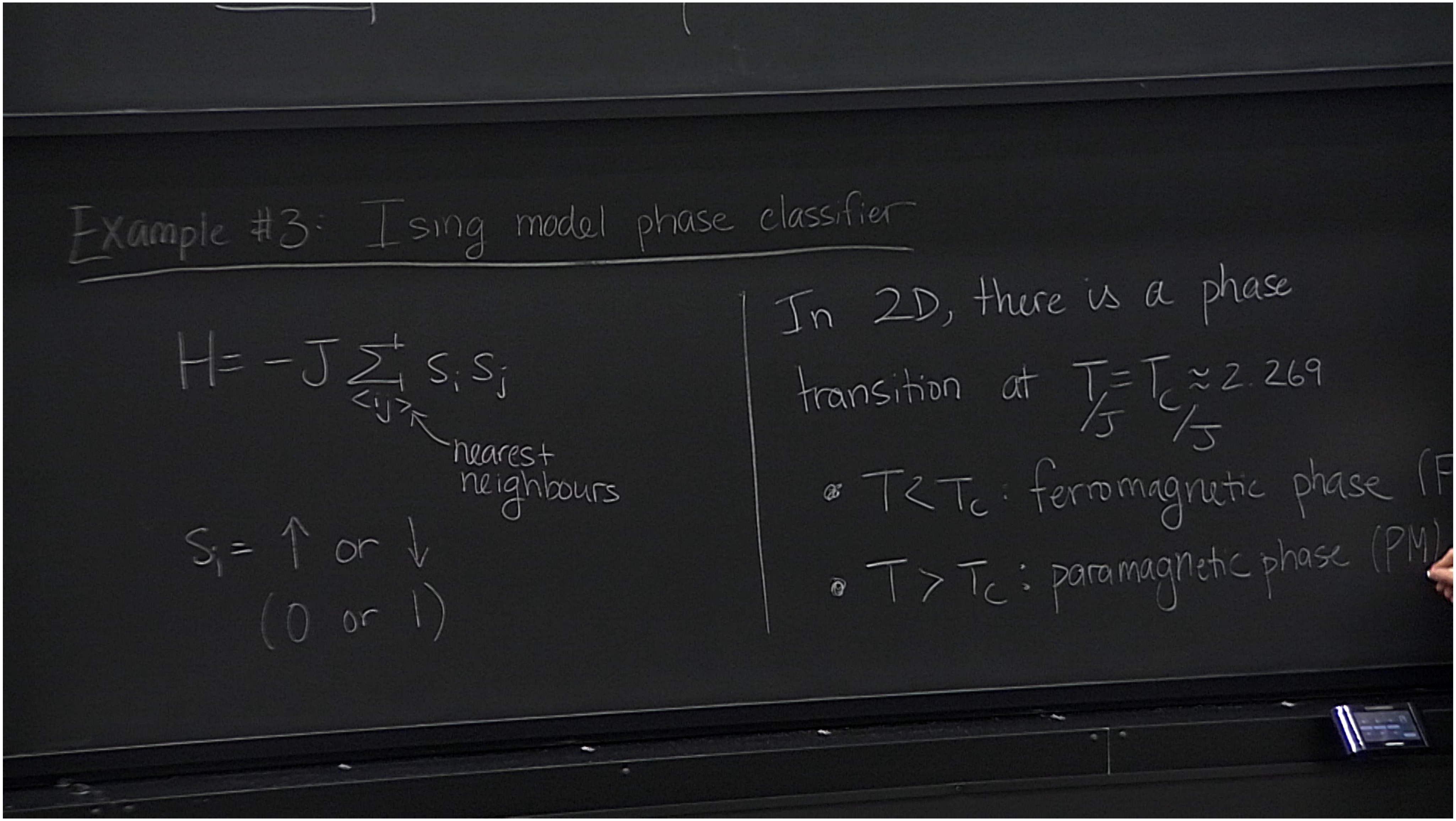
Example #1: 1D regression

input data x: 1D real-valued coord.

label y: " " " "


y
x
Goal: determine a
curve to describe
the data


Example #2: Handwritten digits classifier
input data $\vec{x}$: image of a handwritten digit
label y: the corresponding digit (0, 1, ..., 9)
$\vec{x}_1 =$ 9 , $y_1 = 9$
$\vec{x}_2 =$ 1 OR 1 , $y_2 = 1$


Example #3: Ising model phase classifier
H = −J Σ_{<ij>} s_i s_j
nearest neighbours
s_i = ↑ or ↓
(0 or 1)
In 2D, there is a phase
transition at T/J = T_c/J ≈ 2.269
T < T_c: ferromagnetic phase (F
T > T_c: paramagnetic phase (PM)

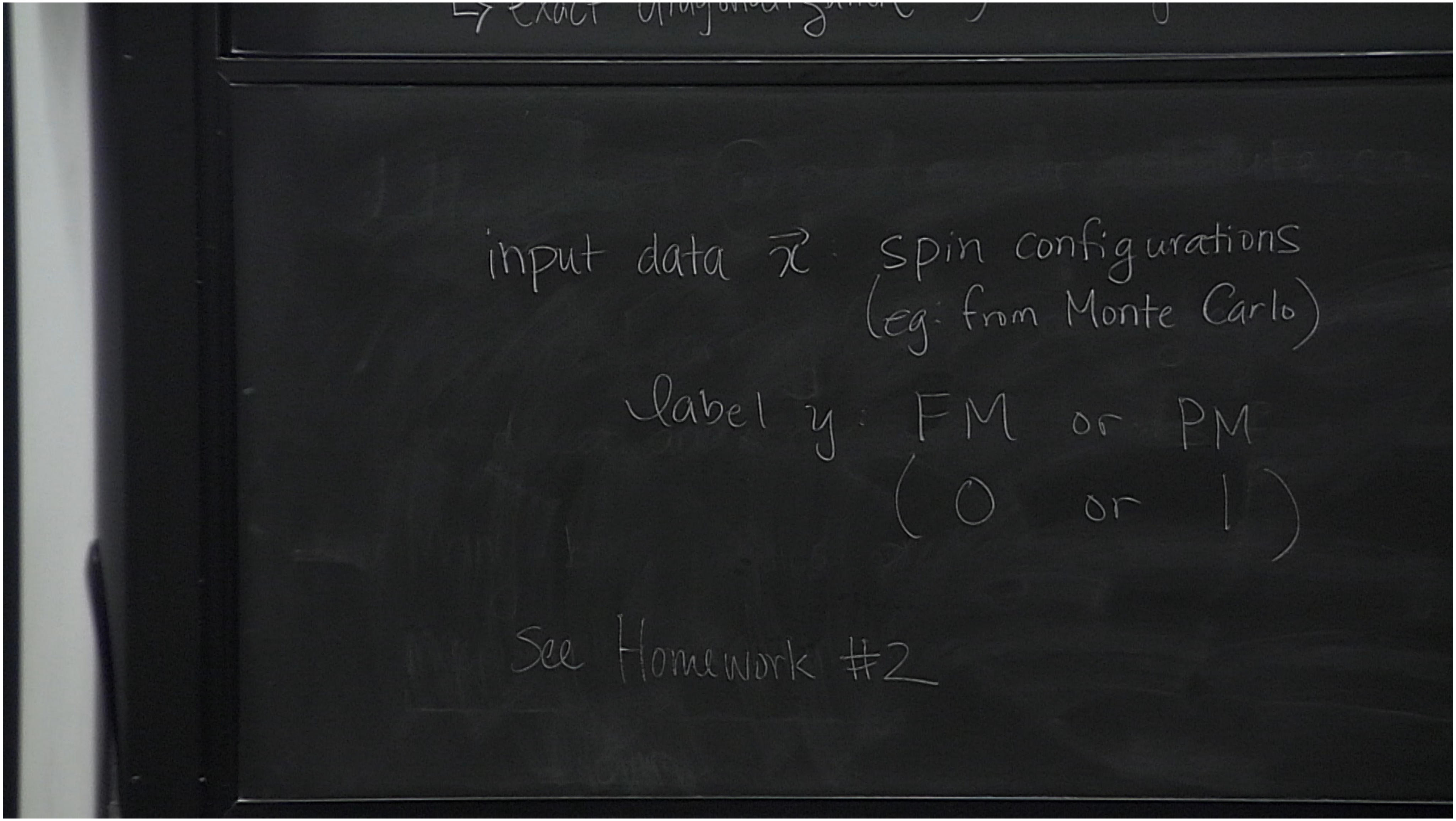
odel phase classifier

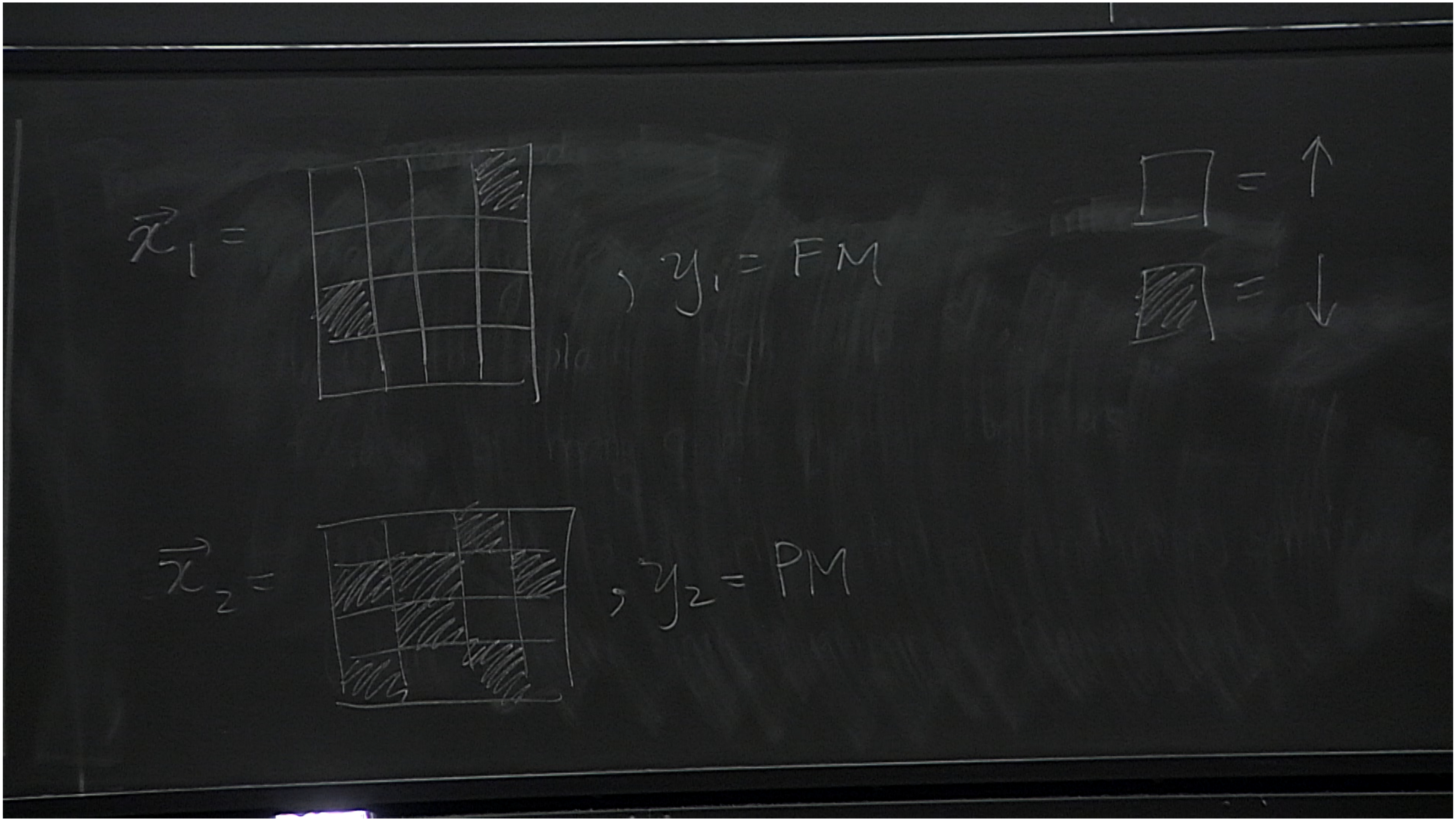
t

bours

In 2D, there is a phase transition at $T/J = T_c/J \approx 2.269$

- $T < T_c$: ferromagnetic phase (FM)
- $T > T_c$: paramagnetic phase (PM)


input data $\vec{x}$: spin configurations
(eg: from Monte Carlo)
label y : FM or PM
(0 or 1)
See Homework #2


$\vec{x}_1 =$
$; y_1 = FM$
$= \uparrow$
$= \downarrow$
$\vec{x}_2 =$
$; y_2 = PM$