

Are Computers Better than Physicists at Physics?

Applications & Limitations of Deep Learning in Physics

Sanha Cheong

sanha@stanford.edu



Department of Physics
Stanford University

July 8, 2018



In this talk, I will cover:

- ▶ **Introduction to deep learning**
 - ▶ **What is deep learning? What are neural networks?**
 - ▶ **What can neural networks do? How?**
- ▶ Deep learning in physics
 - ▶ Data challenges in physics research
 - ▶ How deep learning has helped so far
 - ▶ Limitations of deep learning for scientific research

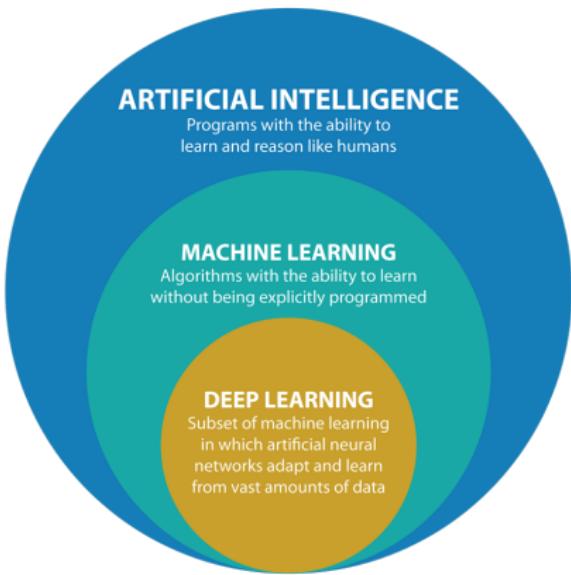


Everyone talks about *buzz-words* like artificial intelligence (AI) and machine learning (ML), but *what do they actually mean?*

Definitions:

- ▶ AI: human-like, *intelligent* machines or programs
- ▶ ML: AI algorithms that *learn from data*

(This is still vague,
but don't worry! We will cover
plenty examples later.)





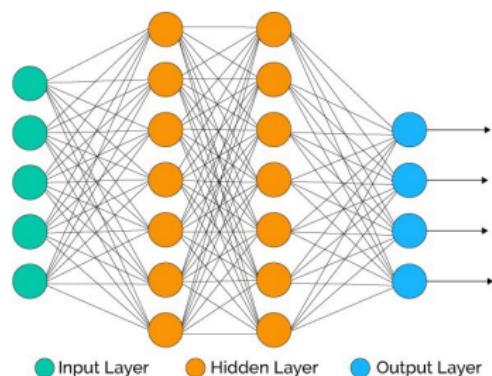
... wait! You missed one: *deep learning!*

Deep learning (DL) is a specific set of ML algorithms that use artificial *neural networks* (NN's)

NN is a *function* whose computational graph mimics the structure of biological neural systems.

NN is defined by:

- ▶ Architecture (comp. graph)
- ▶ Parameters (connections)



Neural Networks

What do NN's actually compute?



Let us zoom into one hidden layer of a simple NN

$$\left[\begin{array}{c} M \text{ of } \\ \begin{array}{c} a_1 \\ a_2 \\ \vdots \\ a_N \end{array} \end{array} \right] \xrightarrow{\begin{array}{c} w_1 \\ w_2 \\ \vdots \\ w_N \end{array}} \begin{array}{c} +1 \\ b \\ \downarrow \\ \Sigma \\ z \\ g \\ a_{out} \end{array} \rightarrow \left[\begin{array}{l} = M \text{ of } g \left(\sum_{i=1}^N w_i a_i + b \right) \\ z = b + \sum_{i=1}^N a_i w_i \\ a_{out} = g(z) \end{array} \right] = g(W\mathbf{a} + \mathbf{b}) \right]$$

where $g : \mathbb{R} \rightarrow \mathbb{R}$ is called the *activation* function, W is the matrix of $M \times N$ *weights*, and \mathbf{b} is the *bias*.

The activation function here is applied element-wise:

$$\mathbf{x} = [x_1, x_2, x_3, \dots, x_N]$$

$$g(\mathbf{x}) = [g(x_1), g(x_2), g(x_3), \dots, g(x_N)]$$



Sure... but so what? This is just a bunch of calculations.

Why are NN's so powerful?

Universal Approximation Theorem

A single-hidden layer NN with sufficient nodes and a non-linear activation function can *approximate any function* with an arbitrary accuracy. NN's are *universal approximators*.

Caveats and details:

- ▶ A single-layer neural network defined by an activation function g and k hidden units $\mathcal{N}(g, k) : \mathbb{R}^{\text{in}} \rightarrow \mathbb{R}^{\text{out}}$ is dense in $L^p(\mu)$ for any finite measure μ
- ▶ The analytical details depend on properties of the activation function g
- ▶ Discussions & proofs mostly limited to continuous functions

Some references:

- ▶ G. Cybenko, "Approximation by Superpositions of a Sigmoidal Function" (1989) [1]
- ▶ K. Hornik, "Approximation Capabilities of Multilayer Feedforward Networks" (1991) [2]

Visualizing Universal Approximation Theorem



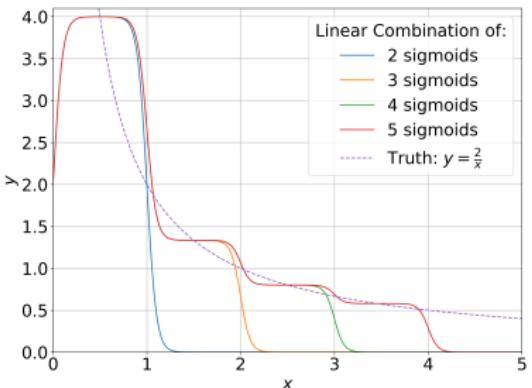
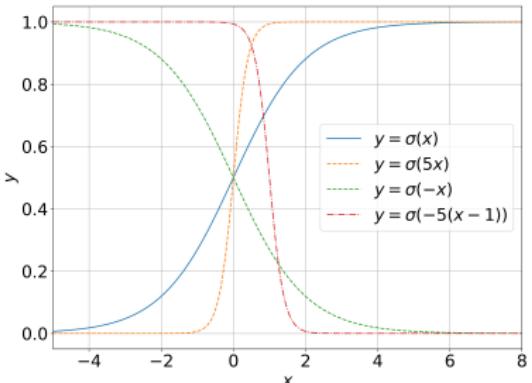
Consider

a NN : $\mathbb{R} \rightarrow \mathbb{R}$ with the *sigmoid* activation: $\sigma(x) \equiv \frac{1}{1+e^{-x}}$.

$$\text{NN}(x) = \sum_{i=1}^k w'_i \sigma(w_i x + b_i) + b'$$

Each sigmoid (non-linearity) can approximate *a local change*

k sigmoids \Rightarrow
approximate at $\sim k$ points



Backpropagation

How NN's learn

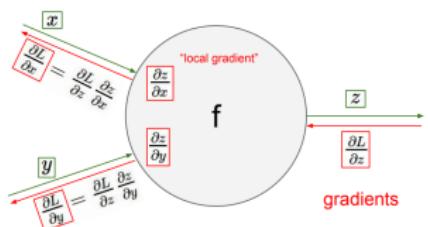


To let the computers learn the right parameters to approximate a function, we use a set of *training dataset* and *loss function*

- ▶ Training dataset: $T = \{(x_i, y_i) | x_i \in X, y_i \in Y\}$
- ▶ Loss function: $\ell(\hat{y}, y)$ measures how *far* a prediction $\hat{y} = \text{NN}(x)$ is from the truth y

“Learning” is now an *optimization problem*; given T , find a set of NN params. that minimize $\ell_{\text{overall}} = \frac{1}{m} \sum_{i=1}^m \ell(\hat{y}_i, y_i)$.

Use *chain rule of derivatives*:



$$\frac{\partial \ell_{\text{overall}}}{\partial h_j} = \frac{\partial \ell_{\text{overall}}}{\partial h_{j+1}} \times \frac{\partial h_{j+1}}{\partial h_j}$$

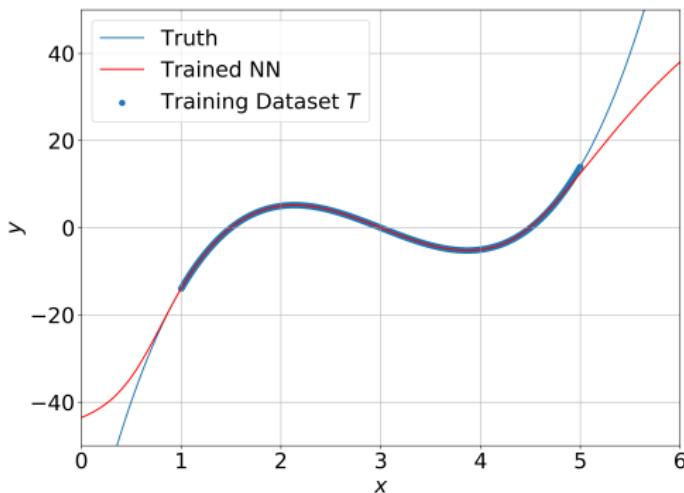
recursively backwards
through the layers of NN

Quick Example

Approximating a polynomial with a NN



- ▶ NN: single hidden layer with $N_{\text{hidden}} = 100$ nodes, σ -activation
- ▶ $T = \{(x_i, y_i) | i = 1, \dots, 50k, 1 \leq x_i \leq 5,$
$$y_i = 4x_i^3 - 36x_i^2 + 99x_i - 81\}$$
- ▶ Loss function: $\ell(\hat{y}, y) = |\hat{y} - y|$



Convolutional Neural Network

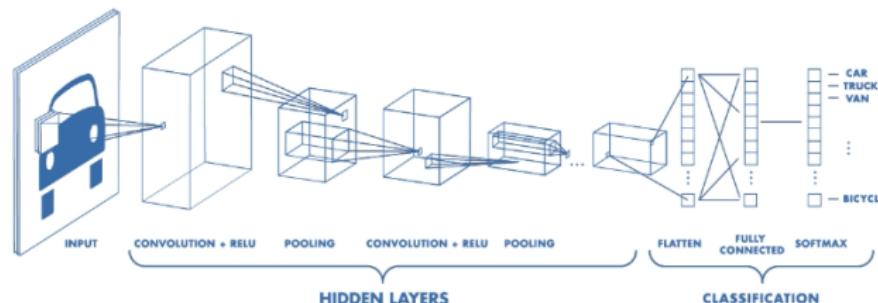
More sophisticated NN's: part 1



Each edge in the computational graph is *not necessarily limited* to matrix multiplications

Convolutional Neural Network (CNN, ConvNet)

- ▶ Striding filters across different axes (\sim convolution integral)
- ▶ Understands *positions / geometry*



- ▶ Successes in image classification and other “vision” tasks

CNN: Successful Examples

Computer vision

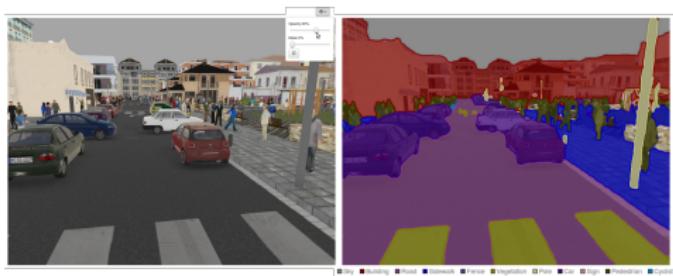


```
probability=0.569505, class=n02088364 beagle  
probability=0.052795, class=n00452864 beagling  
probability=0.039277, class=n02778669 ball  
probability=0.017777, class=n02087122 hunting dog  
probability=0.016321, class=n10611613 sleuth, sleuthhound
```



Detection
(classification + localization)

```
probability=0.692314, class=n02122948 kitten, kitty  
probability=0.043846, class=n01323155 kit  
probability=0.030001, class=n01318894 pet  
probability=0.029692, class=n02122878 tabby, queen  
probability=0.026972, class=n01322221 baby
```



Segmentation (pixel-wise labeling)

Recurrent Neural Network

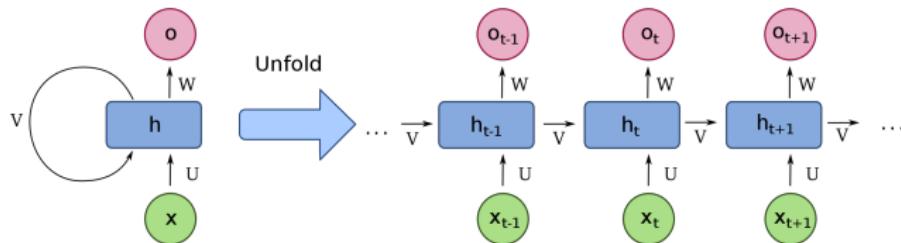
More sophisticated NN's: part 2



Computational graphs *need not be one-directional*

Recurrent Neural Network (RNN)

- ▶ Previous outputs are re-entered as inputs
- ▶ Can deal with sequences of variable lengths
- ▶ Understands *order and context* (has memory)



- ▶ Successes in natural language tasks (translation, semantic understanding) and time-sequential data (future prediction)

RNN: Successful Examples

Natural language processing, time-sequences



Translate

Turn off instant translation

English Spanish French Korean - detected

English Spanish Arabic

Translate

구글 번역기 역시 RNN 을 사용합니다

Google Translator also uses RNN

21/5000

전자항공권 여정 안내서

서울특별시 to 샌프란시스코
매리미니 Flight 212

Jul 21 8:40 PM

Jul 21 3:30 PM

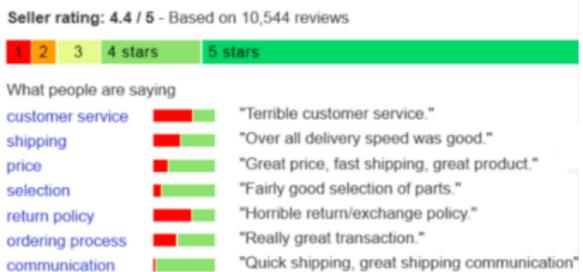
Flight duration
10 hr. 50 min

Confirmation number
ASIANA

ASIANA AIRLINES to me

A STAR ALLIANCE MEMBER

ASIANA AIRLINES



Generative Adversarial Network

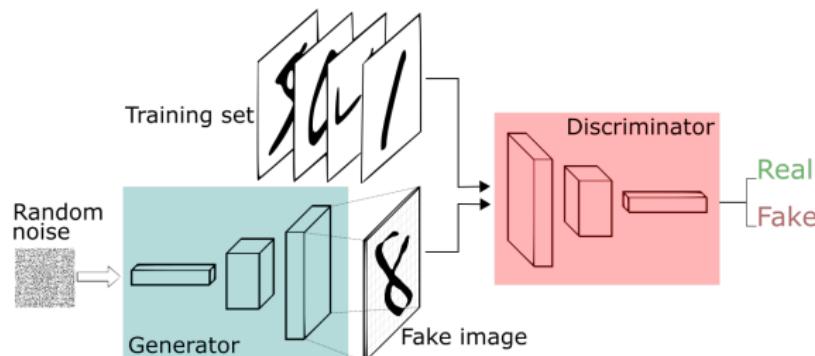
More Sophisticated NN's: Part 3



We can *combine multiple NN's* for more complicated tasks

Generative Adversarial Network (GAN)

- ▶ Two NN's, generator \mathcal{G} and discriminator \mathcal{D} , compete



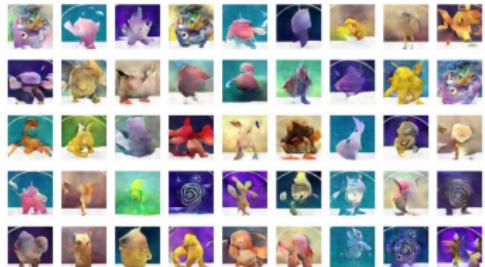
- ▶ \mathcal{G} fools \mathcal{D} , \mathcal{D} distinguishes real data v.s. generated data

$$\min_{\mathcal{G}} \max_{\mathcal{D}} \mathbb{E}_{x \sim p_{\text{data}}(x)} \log \mathcal{D}(x) + \mathbb{E}_{z \sim p_z(z)} \log (1 - \mathcal{D}(\mathcal{G}(z)))$$

- ▶ Eventually, \mathcal{G} becomes good at *generating realistic data*

GAN: Successful Examples

Generating ‘realistic’ images



bicubic
(21.59dB/0.6423)

SRGAN
(21.15dB/0.6868)

original



Grayscale, original, GAN-colored

a clock tower in the middle of a city



a brown horse standing on top of a dirt field



a brown horse standing in a field of grass

a group of people riding horses on a dirt road



In this talk, I will cover:

- ▶ Introduction to deep learning
 - ▶ What is deep learning? What are neural networks?
 - ▶ What can neural networks do? How?
- ▶ Deep learning in physics
 - ▶ Data challenges in physics research
 - ▶ How deep learning has helped so far
 - ▶ Limitations of deep learning for scientific research

Applying All of These to Physics Research



Some data challenges we covered so far:

In physics research, these are:

- ▶ Fitting a curve to points
 → *regression and parameter estimation*
- ▶ Dog or cat?
 → *Signal or noise? What kind of signal?*
- ▶ Locating humans within image data
 → *Finding signals within the detector*
- ▶ Contextual understanding of a word within a sentence
 → *Understanding time- (and other-) sequential data*
- ▶ Generating and coloring realistic images
 → *Simulating realistic physical processes*

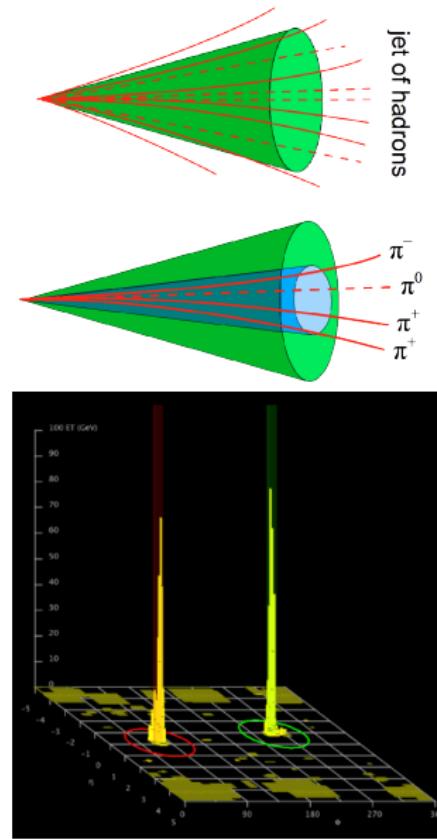
Data Challenges in Particle Physics



Some common data challenges
in *experimental particle physics*:

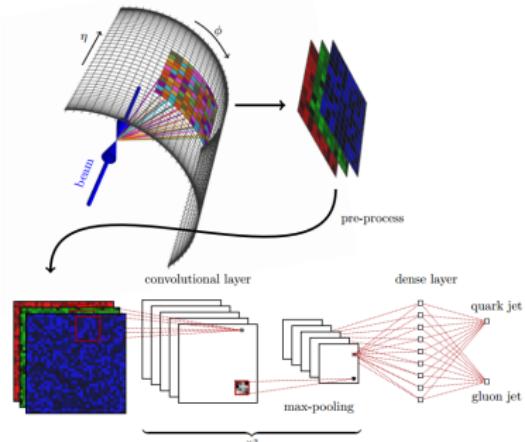
- ▶ Classification: identify the original particle from its decay product
- ▶ De-noising: remove noises from our detector responses
- ▶ Simulation:
mimic collision events @ LHC

In particular, I will mainly
talk about *jet physics* today.



Jet Classification

Quark-gluon tagging and more



"Deep Learning in Color: towards Automated Quark/Gluon Jet Discrimination" (2017)

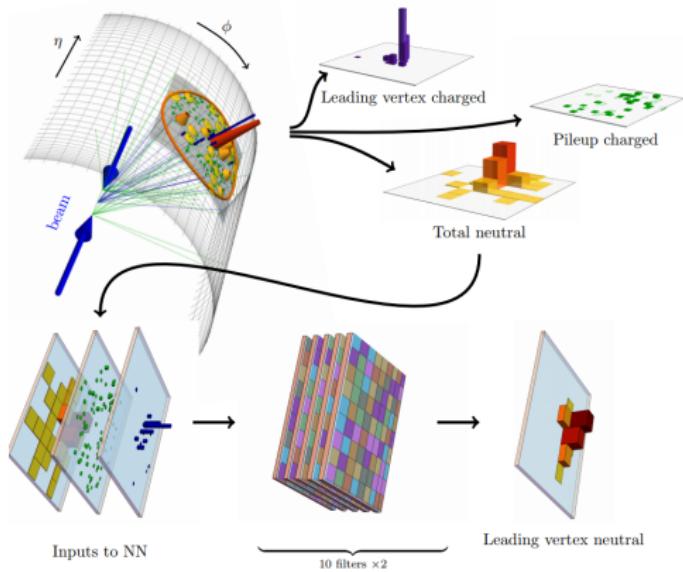
“Unroll”
cylindrical detector
(e.g. ATLAS, CMS)
to form *jet images*

- ▶ Each pixel: direction of particle flight
- ▶ Color channel: features like energy

Use CNN's to *classify*
these images.

Quark v.s. gluon,
flavor-tagging, W^\pm
jets, merged objects

Jet De-noising



"Pileup Mitigation with Machine Learning (PUMML)" (2017)

Similar technique is used (by similar people) to *clean up* jet images and *only leave important variables*

Jet Simulation with GAN's



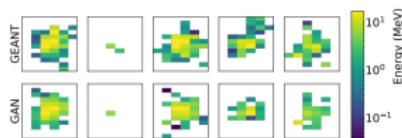
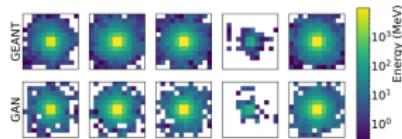
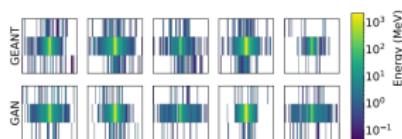
Monte Carlo simulation of jets @ LHC is *extremely* slow

Theory → Collision → Decay (hadronization)
→ Detector simulation, electronics → ... and more

... but GAN's could *accelerate* this!

CaloGAN: Calorimeter GAN

- ▶ Generate full simulations of different physics processes
- ▶ Use energy deposits in 3 layers of calorimeters as “jet images”
- ▶ Train GAN's that generates similar jet images



“CaloGAN” (2017)

Data Challenges in Cosmology



Common data challenges in *observational cosmology*:

- ▶ Parameter estimation: estimate physical parameters like Ω_m and Ω_Λ from cosmic mass distribution
- ▶ Simulation: cosmic-scale general-relativistic fluid dynamics simulation
- ▶ Image processing: extract core info. from telescope images

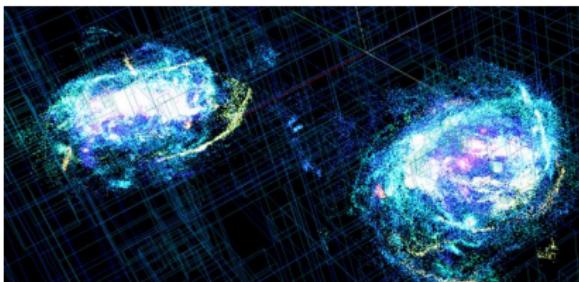
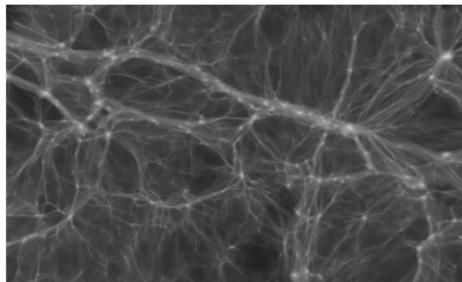


Image credit: Stanford/SLAC KIPAC
Computational Astrophysics & Dark Energy Groups

Cosmological Parameter Estimation

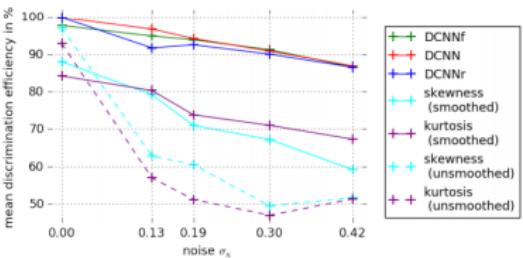
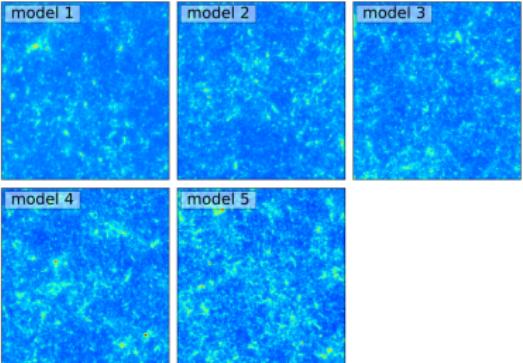
Classifying different, degenerate models



Some cosmological parameters are *degenerate*; different pair of params. give very *similar distributions* in the universe.

- ▶ Run full cosmo. simulations with a set of params
- ▶ Train CNN's on the simulated images

This approach can be generalized into a regression problem; predict the true parameters *hypothesis-free*



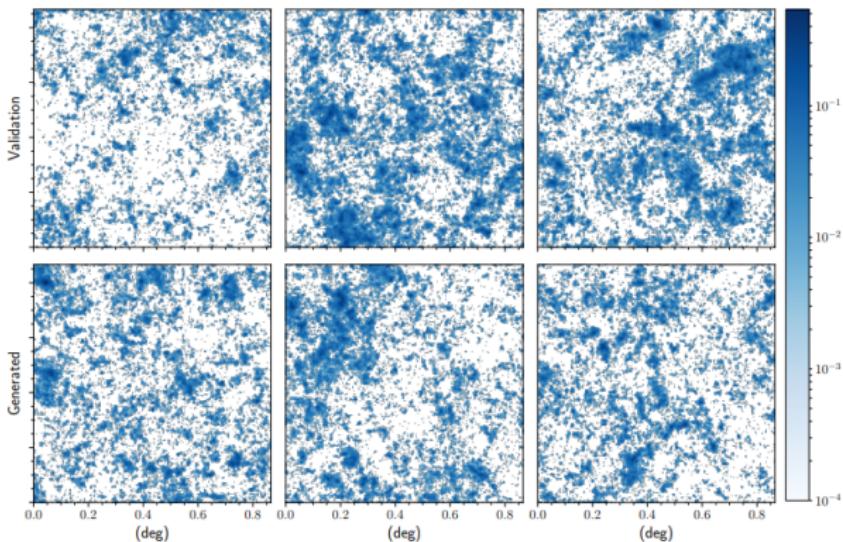
“Cosmological Model Discrimination with Deep Learning” (2017)

Cosmological Simulation with GAN's



*Cosmic-scale general-relativistic
fluid dynamics simulation*

... that already sounds intimidating, but we have GAN's!



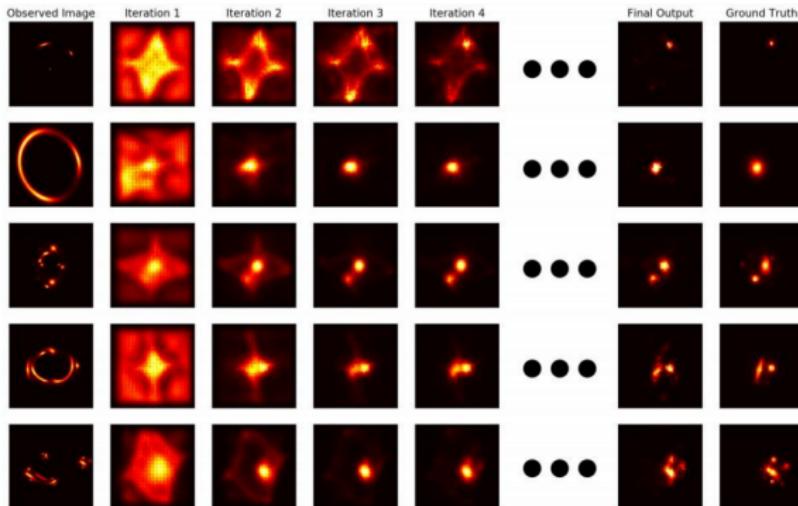
"Creating Virtual Universes Using Generative Adversarial Networks" (2017)

Inverting Gravitational Lensing Effects

Recurrent Inference Machines



*Gravitational lenses distort images of galaxies and other astronomical objects, and inverting distortions is **extremely difficult**.*



“MagNet: Deep Machine Vision for the Cosmic Dawn” (2018)

Limitations of Deep Learning in Scientific Research



Deep learning algorithms seem amazing so far, but:

- ▶ Difficult to quantify their *uncertainties*
 - ▶ No clear rule for error propagation
 - ▶ *Understanding and calibrating* NN outputs
 - ▶ *in situ* methods are expensive
- ▶ They don't *understand* physics
 - ▶ In examples seen so far, they *hint at* previously unknown physics, at best
 - ▶ Physicists have not only data-driven, but also *subjective* measures of “good physics”
- ▶ There are *unquantifiable* physics problems to solve
 - ▶ Hierarchy problem, naturalness problem
 - ▶ Looking for “elegant” theories

Thank you for your attention!

and I'm happy to take any questions now or later :)



- [1] G. Cybenko.
Approximation by superpositions of a sigmoidal function.
Mathematics of Control, Signals, and Systems (MCSS),
2(4):303–314, December 1989.

- [2] Kurt Hornik.
Approximation capabilities of multilayer feedforward networks.
Neural Netw., 4(2):251–257, March 1991.



- [3] The Apache Software Foundation.

Incubator mxnet: Image classification example.

[https://github.com/apache/incubator-mxnet/
tree/master/example/image-classification](https://github.com/apache/incubator-mxnet/tree/master/example/image-classification),
2018.

- [4] S. K. Divvala, D. Hoiem, J. H. Hays, A. A. Efros, and
M. Hebert.

An empirical study of context in object detection.

In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1271–1278, June 2009.



Reference (cont.)

- [5] G. Ros, L. Sellart, J. Materzynska, D. Vazquez, and A. M. Lopez.

The synthia dataset: A large collection of synthetic images for semantic segmentation of urban scenes.

In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3234–3243, June 2016.

- [6] Ouais Alsharif, Tom Ouyang, Francoise Beaufays, Shumin Zhai, Thomas Breuel, and Johan Schalkwyk.

Long short term memory neural network for keyboard gesture decoding, 04 2015.



Reference (cont.)

- [7] Christian Ledig, Lucas Theis, Ferenc Huszar, Jose Caballero, Andrew P. Aitken, Alykhan Tejani, Johannes Totz, Zehan Wang, and Wenzhe Shi.
Photo-realistic single image super-resolution using a generative adversarial network.
CoRR, abs/1609.04802, 2016.
- [8] Kevin Frans.
Outline colorization through tandem adversarial networks.
CoRR, abs/1704.08834, 2017.



- [9] Kamyar Nazeri and Eric Ng.
Image colorization with generative adversarial networks.
CoRR, abs/1803.05400, 2018.
- [10] Scott E. Reed, Zeynep Akata, Xinchen Yan, Lajanugen Logeswaran, Bernt Schiele, and Honglak Lee.
Generative adversarial text to image synthesis.
CoRR, abs/1605.05396, 2016.
- [11] Jiale Zhi.
Pixelbrush: Art generation from text with gans.
<http://cs231n.stanford.edu/reports/2017/pdfs/322.pdf>, 2017.



- [12] Yongcheng Jing, Yezhou Yang, Zunlei Feng, Jingwen Ye, and Mingli Song.

Neural style transfer: A review.

CoRR, abs/1705.04058, 2017.

- [13] Patrick T. Komiske, Eric M. Metodiev, and Matthew D. Schwartz.

Deep learning in color: towards automated quark/gluon jet discrimination.

JHEP, 01:110, 2017.



- [14] Patrick T. Komiske, Eric Mario Metodiev, Benjamin Nachman, and Matthew D. Schwartz.

Pileup mitigation with machine learning (pumml).

2017, 07 2017.

- [15] M. Paganini, L. de Oliveira, and B. Nachman.

Accelerating Science with Generative Adversarial Networks:
An Application to 3D Particle Showers in Multilayer
Calorimeters.

Physical Review Letters, 120(4):042003, January 2018.



- [16] Jorit Schmelzle, Aurelien Lucchi, Tomasz Kacprzak, Adam Amara, Raphael Sgier, Alexandre Réfrégier, and Thomas Hofmann.

Cosmological model discrimination with Deep Learning.

2017.

- [17] Mustafa Mustafa, Deborah Bard, Wahid Bhimji, Rami Al-Rfou, and Zarija Lukić.

Creating virtual universes using generative adversarial networks.

06 2017.



[18] Warren Morningstar and Dylan Rueter.

Magnet: Gravitational lens source reconstruction using deep learning.

http://cs230.stanford.edu/files_winter_2018/projects/6925320.pdf, 2018.