Introduction to Deep Learning for Mathematicians by a Physicist

Capabilities of Neural Networks: Matheamtical and Empirical Perspectives

Sanha Cheong
sanha@stanford.edu

Department of Physics
Stanford University

July 16, 2018

Agenda

In this talk, I will cover:

Agenda

In this talk, I will cover:

- Introduction to (supervised) deep learning
  - What is deep learning? What are neural networks?
  - What can neural networks do? How?

Agenda

In this talk, I will cover:

- Introduction to (supervised) deep learning
  - What is deep learning? What are neural networks?
  - What can neural networks do? How?
- Math. perspective on capabilities of neural networks
  - Universal Approximation Theorem and its proof
  - Visualizing what neural networks can do

Agenda

In this talk, I will cover:

- Introduction to (supervised) deep learning
  - What is deep learning? What are neural networks?
  - What can neural networks do? How?
- Math. perspective on capabilities of neural networks
  - Universal Approximation Theorem and its proof
  - Visualizing what neural networks can do
- Empirical perspective on capabilities of neural networks
  - More practical and sophisticated neural networks and their success in human tasks
  - Deep learning in physics research

Agenda

In this talk, I will cover:

- **Introduction to (supervised) deep learning**
  - **What is deep learning? What are neural networks?**
  - **What can neural networks do? How?**
- Math. perspective on capabilities of neural networks
  - Universal Approximation Theorem and its proof
  - Visualizing what neural networks can do
- Empirical perspective on capabilities of neural networks
  - More practical and sophisticated neural networks and their success in human tasks
  - Deep learning in physics research

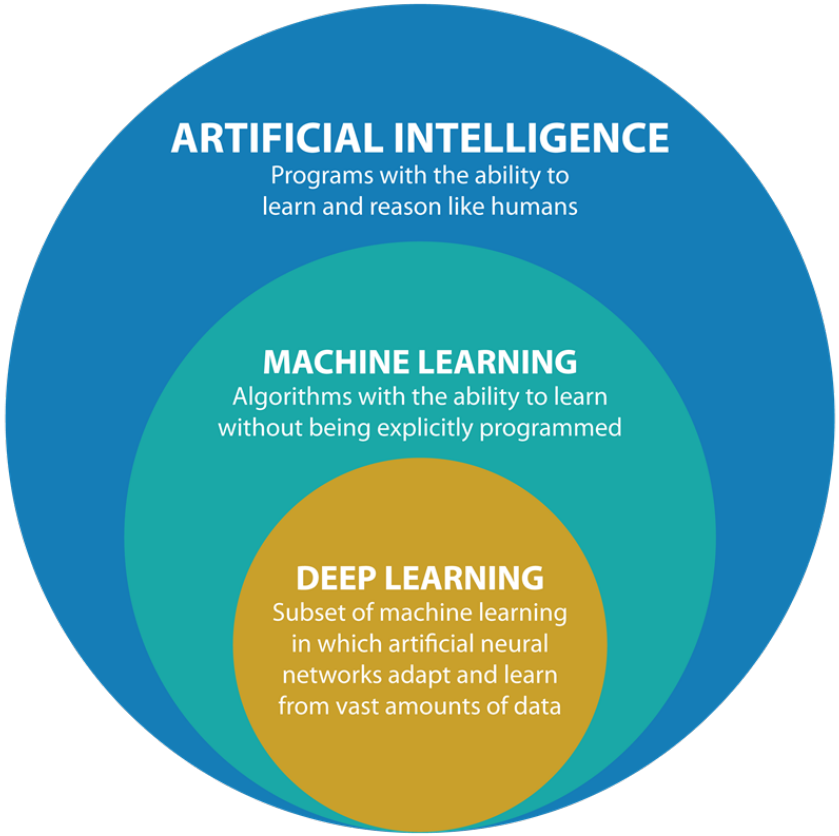
Everyone talks about *buzz-words* like artificial intelligence (AI) and machine learning (ML), but *what do they actually mean*?

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)

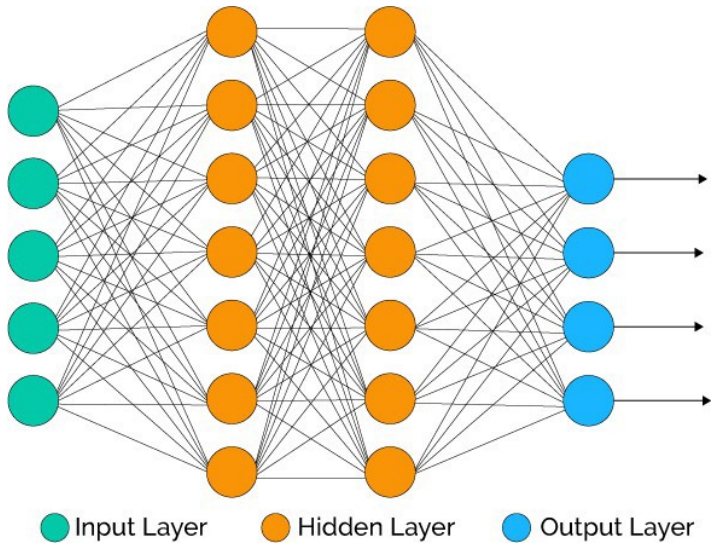
... wait! You missed one: *deep learning!*

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

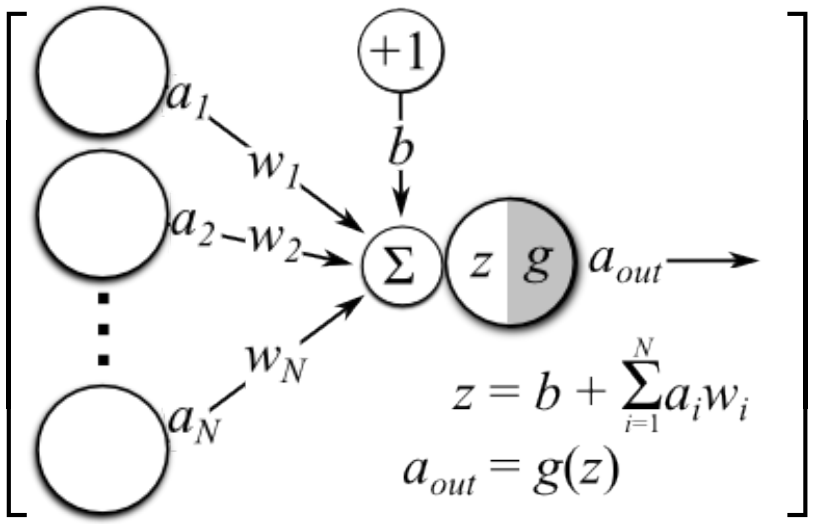
NN is a *function* whose computational graph mimics the structure of biological neural systems.

NN is defined by:

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



Let us zoom into one hidden layer of a simple NN

M of 

$$= M \text{ of } g\left(\sum_{i=1}^{N} w_i a_i + b\right)$$

$$= g\left(W\mathbf{a} + \mathbf{b}\right)$$

where $g : \mathbb{R} \to \mathbb{R}$ 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:

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

Agenda

In this talk, I will cover:

- Introduction to (supervised) deep learning
  - What is deep learning? What are neural networks?
  - What can neural networks do? How?
- **Math. perspective on capabilities of neural networks**
  - **Universal Approximation Theorem and its proof**
  - **Visualizing what neural networks can do**
- Empirical perspective on capabilities of neural networks
  - More practical and sophisticated neural networks and their success in human tasks
  - Deep learning in physics research

Sure... but so what? Isn't NN just a bunch of affine transformations & non-linearities?

Why are NN's so powerful?

Sure... but so what? Isn't NN just a bunch of affine transformations & non-linearities?

Why are NN's so powerful?

Universal Approximation Theorem (UAT)

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*.

Sure... but so what? Isn't NN just a bunch of affine transformations & non-linearities?

Why are NN's so powerful?

Universal Approximation Theorem (UAT)

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*.

Mathematically, this is not very precise...

The first version of UAT is given in G. Cybenko, "Approximation by Superpositions of a Sigmoidal Function" (1989) [1]

Cybenko's UAT (1989)

Consider the set $\mathcal{N}$ of all single-hidden layer neural networks $\nu : I_n = [0,1]^n \to \mathbb{R}$ with h hidden nodes:

$$\nu(x) = \sum_{j=1}^{h} c_j \sigma(a_j^T x + b_j)$$

where $a_j \in \mathbb{R}^n, b_j, c_j \in \mathbb{R}$, and the activation function $\sigma : \mathbb{R} \to \mathbb{R}$ is continuous and sigmoidal.

$$\mathcal{N} \text{ is dense in } C(I_n)$$

- $M(I_n) \equiv$ set of all finite, signed regular Borel measures on I_n
- Supremum norm of $f : X \to Y$: $\|f\| \equiv \sup\{|f(x)| : x \in A\}$
- A function $\sigma : \mathbb{R} \to \mathbb{R}$ is *sigmoidal* iff:

$$\lim_{t \to \infty} \sigma(t) = 1 \quad , \quad \lim_{t \to -\infty} \sigma(t) = 0$$

  $\sigma(t) = 1/\left(1 + e^{-t}\right)$ is called *the sigmoid* function.

- A function $f : \mathbb{R} \to \mathbb{R}$ is *discriminatory* iff,
  for some $\mu \in M(I_n)$ and all $a \in \mathbb{R}^n, b \in \mathbb{R}$, we have:

$$\int_{I_n} f\left(a^T x + c\right) \mathrm{d}\mu(x) = 0 \quad \Longrightarrow \quad \mu = 0$$

  Essentially, discriminatory f is volumetrically *non-destructive*.

First, we claim:

continuous sigmoidal functions are discriminatory

For a continuous sigmoidal function σ, we define $\sigma_\alpha(x) \equiv \sigma\left(\alpha(a^T x + b) + \beta\right)$. Then, we have:

$$\lim_{\alpha \to \infty} \sigma_\alpha(x) = \begin{cases} 1 & \text{for } a^T x + b > 0 \\ 0 & \text{for } a^T x + b < 0 \\ \sigma(\beta) & \text{for } a^T x + b = 0 \end{cases}$$

In other words, as $\alpha \to \infty$, $\sigma_\alpha(x)$ converge pointwise and boundedly to:

$$\tau(x) = \begin{cases} 1 & \text{for } a^T x + b > 0 \\ 0 & \text{for } a^T x + b < 0 \\ \sigma(\beta) & \text{for } a^T x + b = 0 \end{cases}$$

Define the hyperplane $\Pi_{a,b} \equiv \{x : a^T x + b = 0\}$ and the half-space $H_{a,b} \equiv \{x : a^T x + b > 0\}$ in I_n.

Now, suppose $\int_{I_n} \sigma\left(\alpha(a^T x + b) + \beta\right) \mathrm{d}\mu(x) = 0$ for some $\mu \in M(I_n)$ and all $a \in \mathbb{R}^n, \alpha, \beta, b \in \mathbb{R}$. Then, for all β, a, b, Lebesgue's *bounded convergence theorem* gives:

$$\begin{aligned} 0 &= \lim_{\alpha \to \infty} \int_{I_n} \sigma_\alpha(x) \, \mathrm{d}\mu(x) \\ &= \int_{I_n} \tau(x) \, \mathrm{d}\mu(x) \\ &= \sigma(\beta)\mu(\Pi_{a,b}) + \mu(H_{a,b}) \end{aligned}$$

which implies $\mu(H_{a,b}) = 0$ *for any* a, b.

Consider any given $a \in \mathbb{R}^n$. For any bounded measurable function χ, define the linear functional:

$$F(\chi) = \int_{I_n} \chi(a^T x)\, \mathrm{d}\mu(x)$$

If χ is the indicator function of $[b, \infty)$, then:

$$F(\chi) = \mu(\Pi_{a,-b}) + \mu(H_{a,-b}) = 0$$

for any b, and the same holds for the open interval (b, ∞).

Since functionals of this form are linear, $F(\chi) = 0$ for all simple functions, which are dense in $L^\infty(\mathbb{R})$. *Therefore,* $F = 0$.

Now, note that $u(x) = \exp\left(ia^T x\right)$ is bounded and measurable. Then:

$$F(u) = \int_{I_n} \exp\left(ia^T x\right) \mathrm{d}\mu(x) = 0$$

for all $a \in \mathbb{R}^n$. That is, the Fourier transform of μ is zero, so μ *must be zero* as well.

Thus, *any continuous sigmoidal function σ is discriminatory*.

Suppose the closure $\overline{\mathcal{N}} \neq C(I_n)$. Since $\overline{\mathcal{N}}$ is a linear subspace in $C(I_n)$, *Hahn-Banach theorem* tells us that there exists a bounded linear functional $L : C(I_n) \to \mathbb{R}$ such that $L \neq 0$, but $L(\mathcal{N}) = L(\overline{\mathcal{N}}) = 0$.

Riesz–Markov–Kakutani representation theorem says that any bounded linear functional on $C(I_n)$ can be represented as an integration w.r.t. a unique regular Borel measure. More precisely, there is a unique regular Borel measure $\mu \in M(I_n)$ such that:

$$L(f) = \int_{I_n} f(x) \, \mathrm{d}\mu(x)$$

for all $f \in C(I_n)$.

Recall that we chose L such that: $L(\overline{\mathcal{N}}) = 0$. In particular, this implies:

$$\int_{I_n} \sigma(a^T x + b)\, \mathrm{d}\mu(x) = 0$$

for all a, b.

Since σ is discriminatory, this implies that $\mu = 0$, which in turn implies that: $L(f) = \int_{I_n} f(x)\, \mathrm{d}\mu(x)$ is identically zero. *This contradicts our construction by Hahn-Banach theorem.*

Hence, the closure $\overline{\mathcal{N}}$ is indeed all of $C(I_n)$.

Thus, the subspace of neural networks $\mathcal{N}$ is dense in $C(I_n)$. Then, for any $f \in C(I_n)$, there is a neural network $\nu \in \mathcal{N}$ that can approximate f with an arbitrary accuracy on I_n.

Details & Caveats about UAT

Details & Caveats about UAT

- Rigorous discussions limited to continuous functions, but *this suffices* for practical/numerical purposes

Details & Caveats about UAT

- Rigorous discussions limited to continuous functions, but *this suffices* for practical/numerical purposes
- K. Hornik, "Approximation Capabilities of Multilayer Feedforward Networks" (1991) [2]
  - Significantly weakens the constraint
  - Any bounded non-constant activation gets the job done
  - The approximating power comes from *the structure of NN's*, not the analytical details of activations

Details & Caveats about UAT

- Rigorous discussions limited to continuous functions, but *this suffices* for practical/numerical purposes
- K. Hornik, "Approximation Capabilities of Multilayer Feedforward Networks" (1991) [2]
  - Significantly weakens the constraint
  - Any bounded non-constant activation gets the job done
  - The approximating power comes from *the structure of NN's*, not the analytical details of activations
- Recent proof with more general, unbounded activations [3]

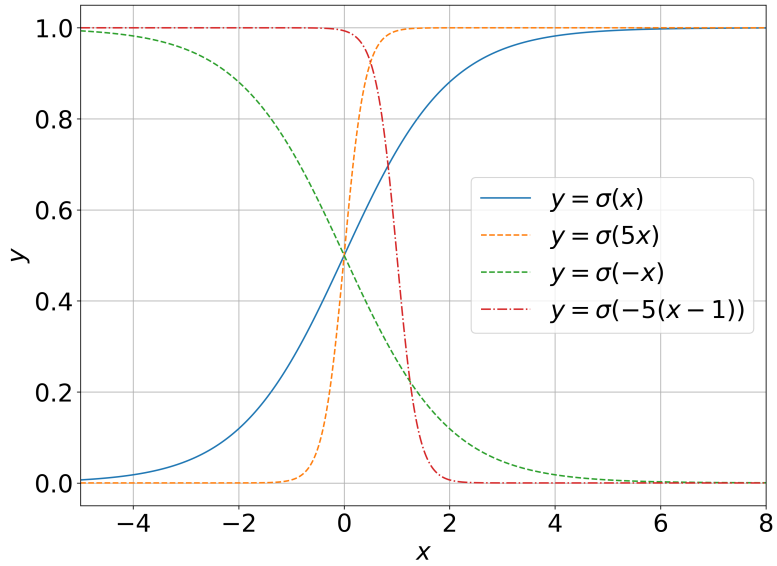
Details & Caveats about UAT

- Rigorous discussions limited to continuous functions, but *this suffices* for practical/numerical purposes
- K. Hornik, "Approximation Capabilities of Multilayer Feedforward Networks" (1991) [2]
  - Significantly weakens the constraint
  - Any bounded non-constant activation gets the job done
  - The approximating power comes from *the structure of NN's*, not the analytical details of activations
- Recent proof with more general, unbounded activations [3]
- Recently popular activation functions include: $\text{ReLU}(x) = \max(0, x)$, $\tanh(x)$, etc.

Visualizing Universal Approximation Theorem

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

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



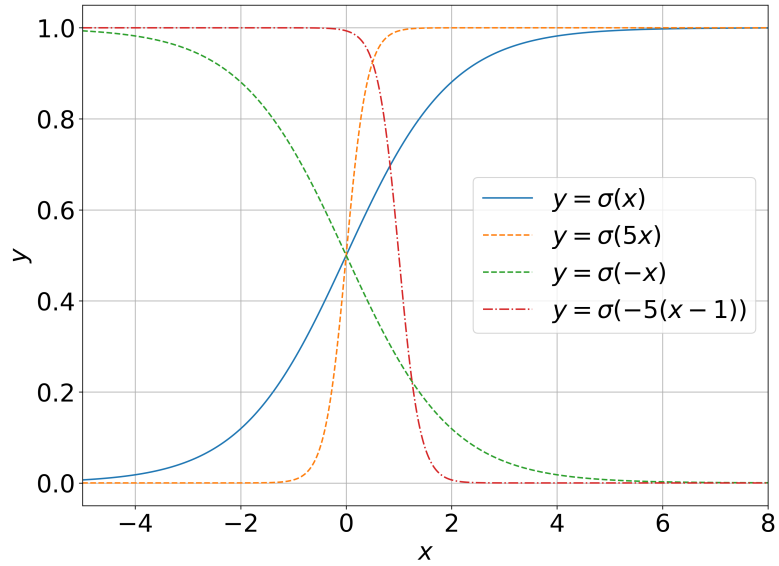
Visualizing Universal Approximation Theorem

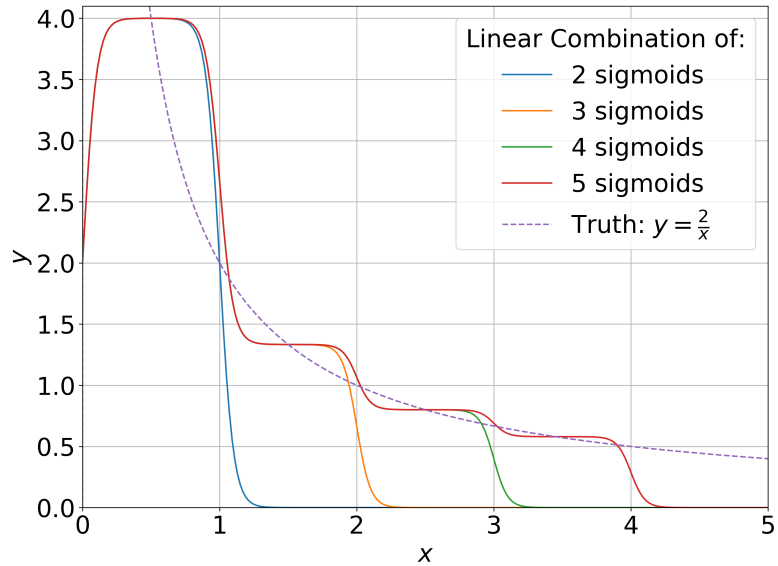
Consider
a NN : $\mathbb{R} \to \mathbb{R}$ with the *sigmoid* activation: $\sigma(x) \equiv \frac{1}{1+e^{-x}}$.

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

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

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





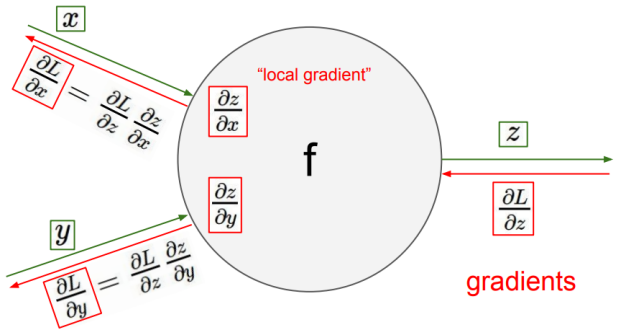
To let the computers learn the right parameters to approximate a function, we use a set of *training dataset* and *loss function*

- 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 $\hat{y} = NN(x)$ is from the truth y

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 $\hat{y} = NN(x)$ is from the truth y

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



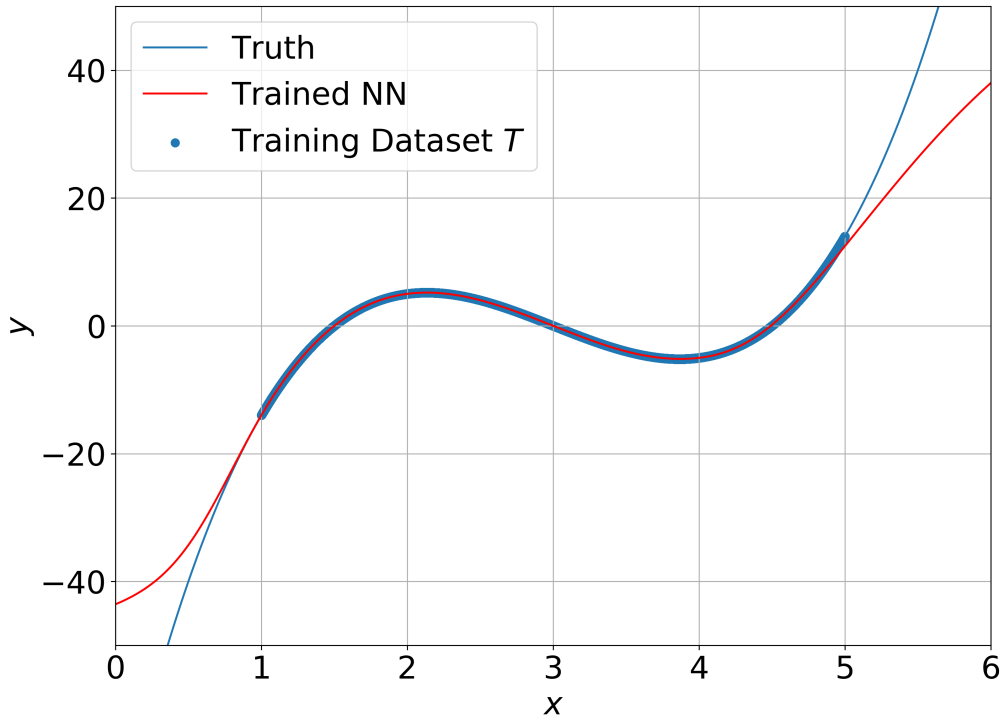
Use *chain rule of derivatives*:

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

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
- $T = \{(x_i, y_i) | i = 1, \ldots, 50\text{k}, 1 \leq x_i \leq 5,$
$$y_i = 4x_i^3 - 36x_i^2 + 99x_i - 81\}$$
- Loss function: $\ell(\hat{y}, y) = |\hat{y} - y|$



Better Examples
Written by others

Just to have some fun:

- Visualize the decision boundary of a NN binary classifier: URL
- Simple NN training live on web: URL

Sure, we've proven UAT, but:

Sure, we've proven UAT, but:

- We don't have infinite training data

Sure, we've proven UAT, but:

- We don't have infinite training data

$$\Rightarrow \text{ generalizable NN's}$$

Sure, we've proven UAT, but:

- We don't have infinite training data

$$\Rightarrow \text{ generalizable NN's}$$

- We don't have infinite computing resources

Sure, we've proven UAT, but:

- We don't have infinite training data

  $\Rightarrow$ generalizable NN's

- We don't have infinite computing resources

  $\Rightarrow$ More sophisticated NN structures

Sure, we've proven UAT, but:

- We don't have infinite training data

  $\Rightarrow$ generalizable NN's

- We don't have infinite computing resources

  $\Rightarrow$ More sophisticated NN structures

  $\Rightarrow$ Better optimization techniques

Sure, we've proven UAT, but:

- We don't have infinite training data

  $\Rightarrow$ generalizable NN's

- We don't have infinite computing resources

  $\Rightarrow$ More sophisticated NN structures

  $\Rightarrow$ Better optimization techniques

  $\Rightarrow$ Parallel computing and GPU research

Sure, we've proven UAT, but:

- We don't have infinite training data

  $\Rightarrow$ generalizable NN's

- We don't have infinite computing resources

  $\Rightarrow$ More sophisticated NN structures

  $\Rightarrow$ Better optimization techniques

  $\Rightarrow$ Parallel computing and GPU research

Hereon, I will introduce *more sophisticated NN's* and their usage.

Agenda

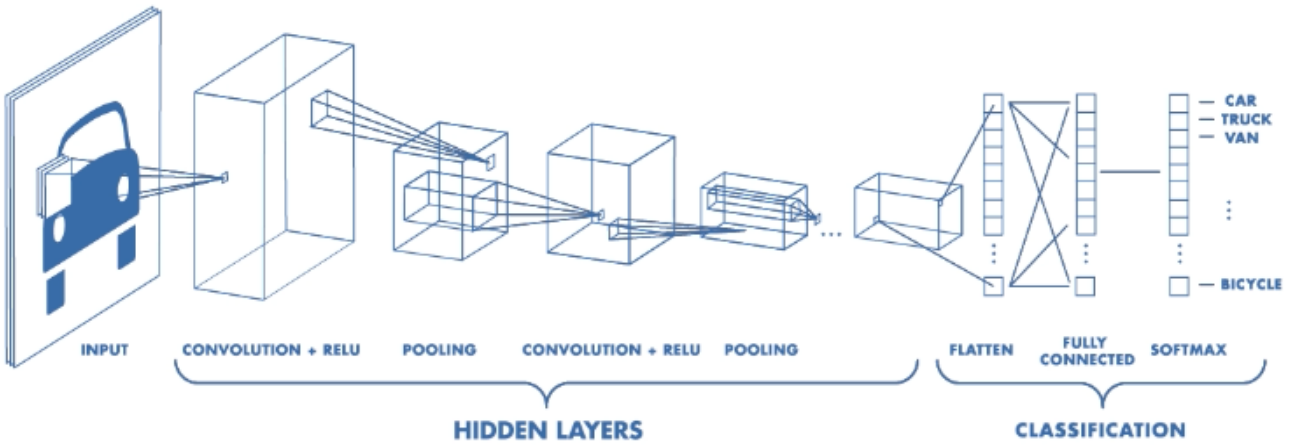
In this talk, I will cover:

- Introduction to (supervised) deep learning
  - What is deep learning? What are neural networks?
  - What can neural networks do? How?
- Math. perspective on capabilities of neural networks
  - Universal Approximation Theorem and its proof
  - Visualizing what neural networks can do
- **Empirical perspective on capabilities of neural networks**
  - **More practical and sophisticated neural networks and their success in human tasks**
  - **Deep learning in physics research**

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*

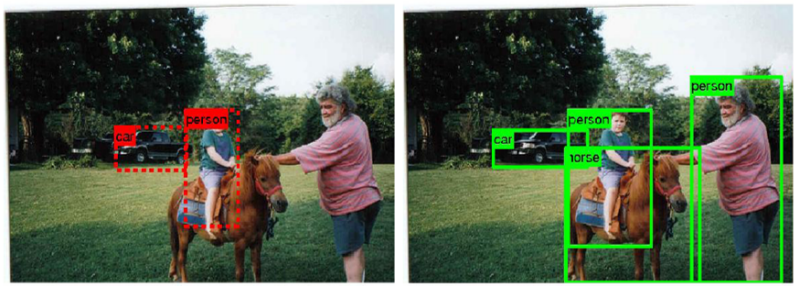


- Successes in image classification and other "vision" tasks

CNN: Successful Examples

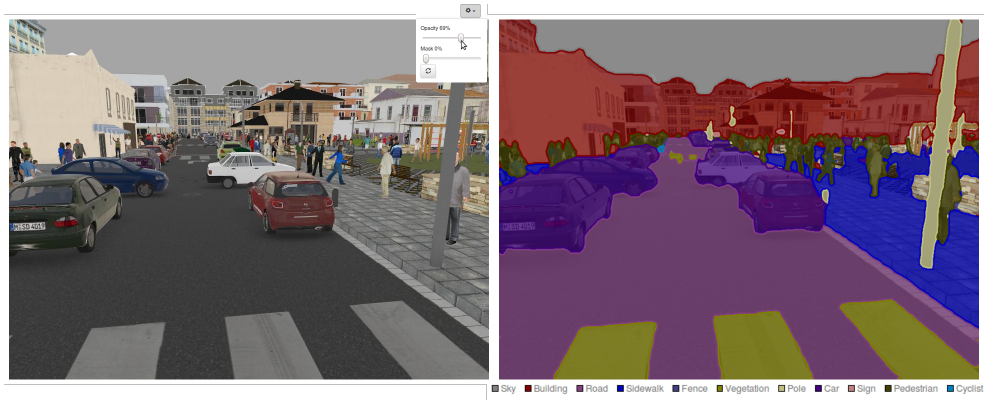
Computer vision

```
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

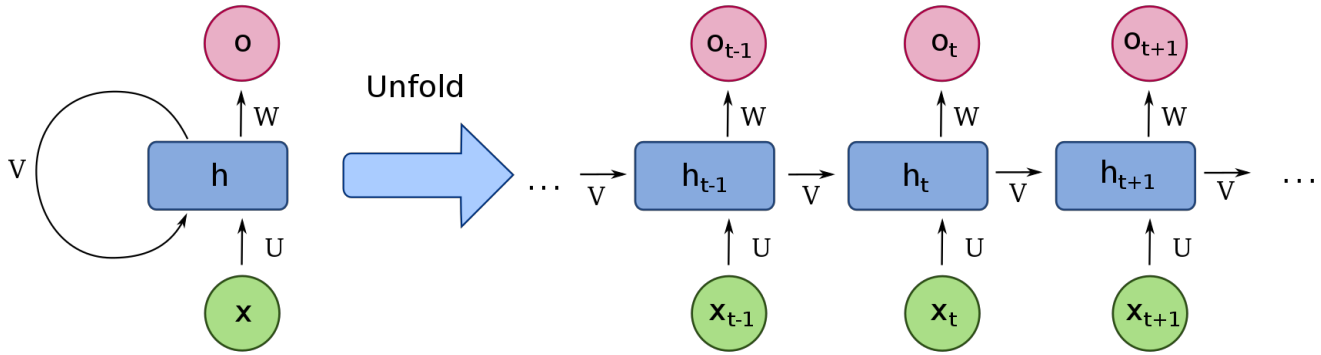


Segmentation (pixel-wise labeling)

Computational graphs *need not be one-directional*

Recurrent Neural Network (RNN)

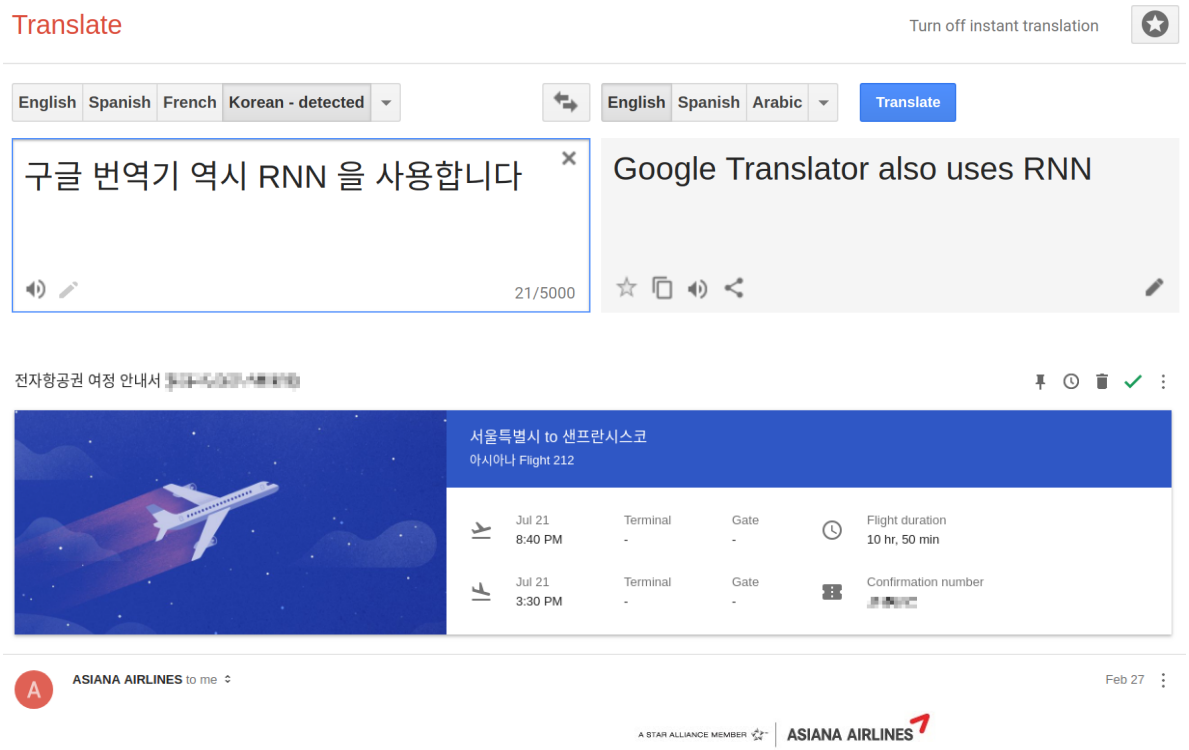
- Previous outputs are re-entered as inputs
- Can deal with sequences of variable lengths
- Understands *order and context* (has memory)

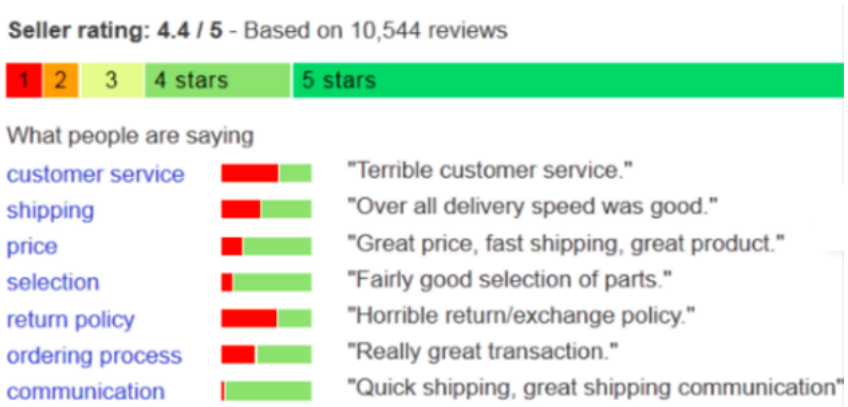


- Successes in natural language tasks (translation, semantic understanding) and time-sequential data (future prediction)

RNN: Successful Examples

Natural language processing, time-sequences

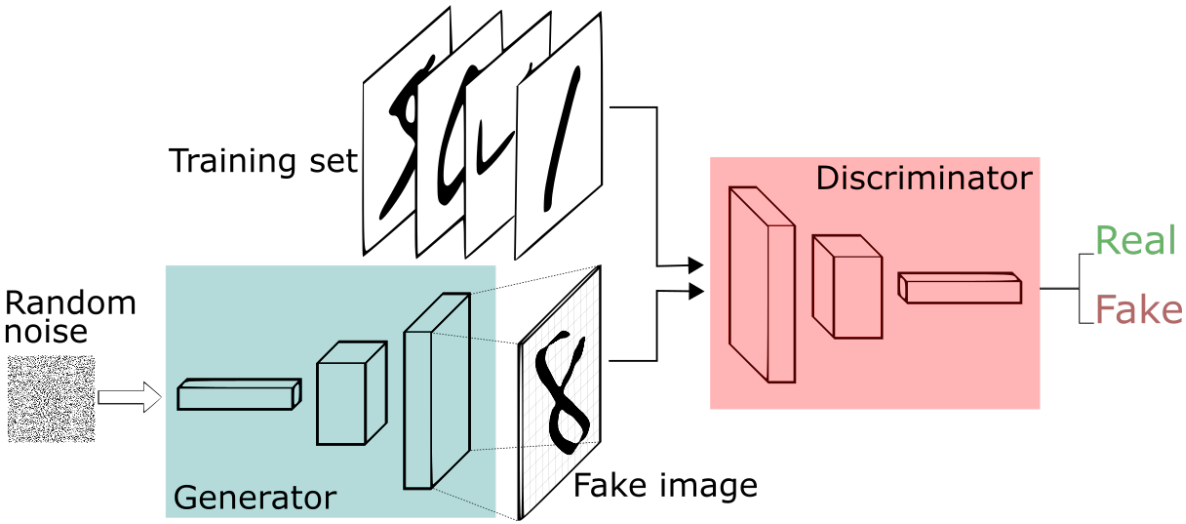




We can *combine multiple NN's* for more complicated tasks

Generative Adversarial Network (GAN)

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



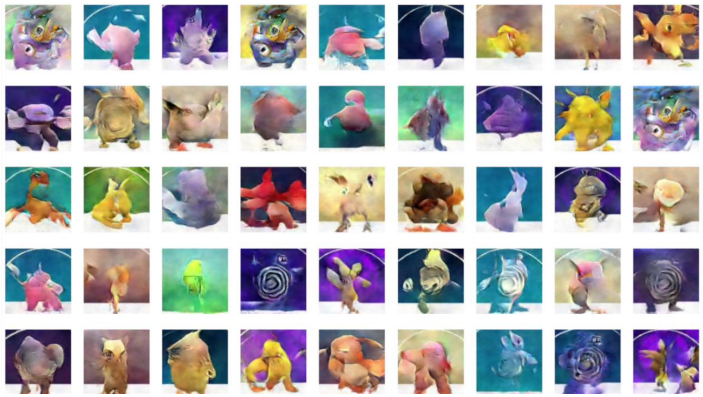
- $\mathcal{G}$ fools $\mathcal{D}$, $\mathcal{D}$ distinguishes real data v.s. generated data

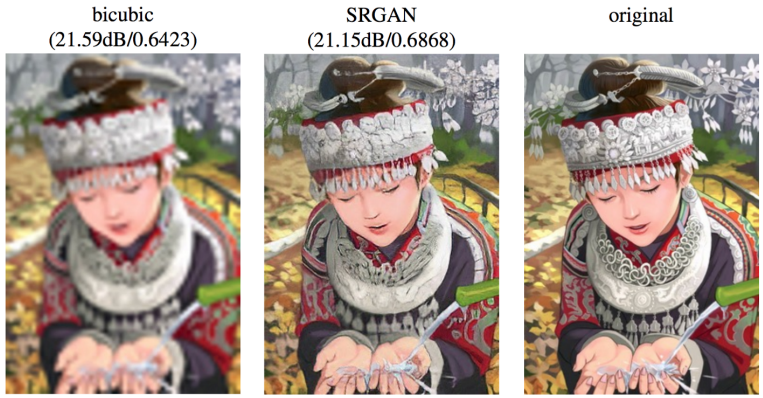
$$\min_{\mathcal{G}} \max_{\mathcal{D}} \mathbb{E}_{x \sim p_{\text{data}}(x)} \log \mathcal{D}(x) + \mathbb{E}_{z \sim p_z(z)} \log \left(1 - \mathcal{D}(\mathcal{G}(z))\right)$$

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

GAN: Successful Examples

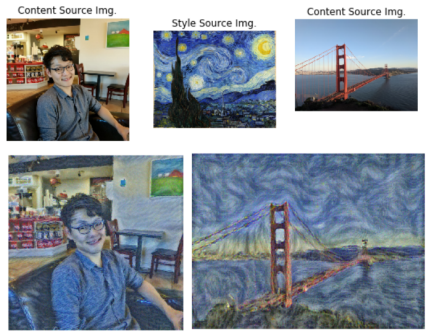
Generating ‘realistic‘ images





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



Applying All of These to Physics Research

Some data challenges we covered so far:

- Fitting a curve to points
- Dog or cat?
- Locating humans within image data
- Contextual understanding of a word within a sentence
- Generating and coloring realistic images

Applying All of These to Physics Research

Some data challenges we covered so far:
In physics research, these are:

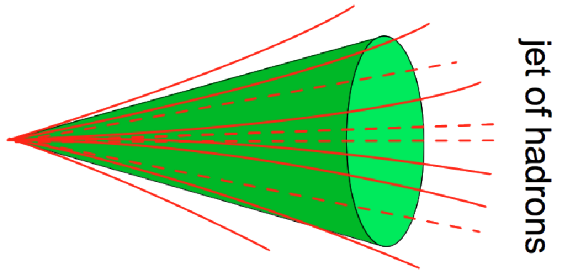
- Fitting a curve to points
  $\longmapsto$ *regression and parameter estimation*
- Dog or cat?
  $\longmapsto$ *Signal or noise? What kind of signal?*
- Locating humans within image data
  $\longmapsto$ *Finding signals within the detector*
- Contextual understanding of a word within a sentence
  $\longmapsto$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

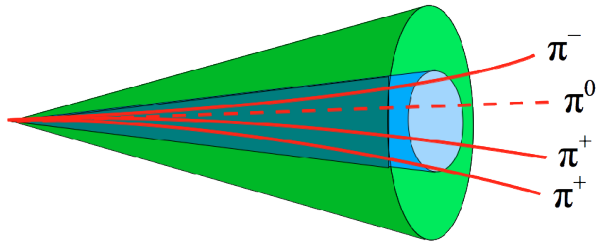
Data Challenges in Particle Physics

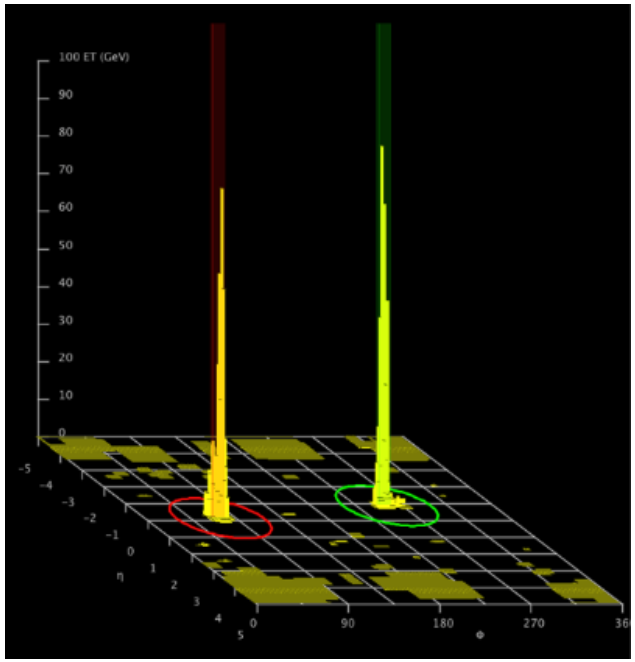
Some common data challenges
in *experimental particle physics*:

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

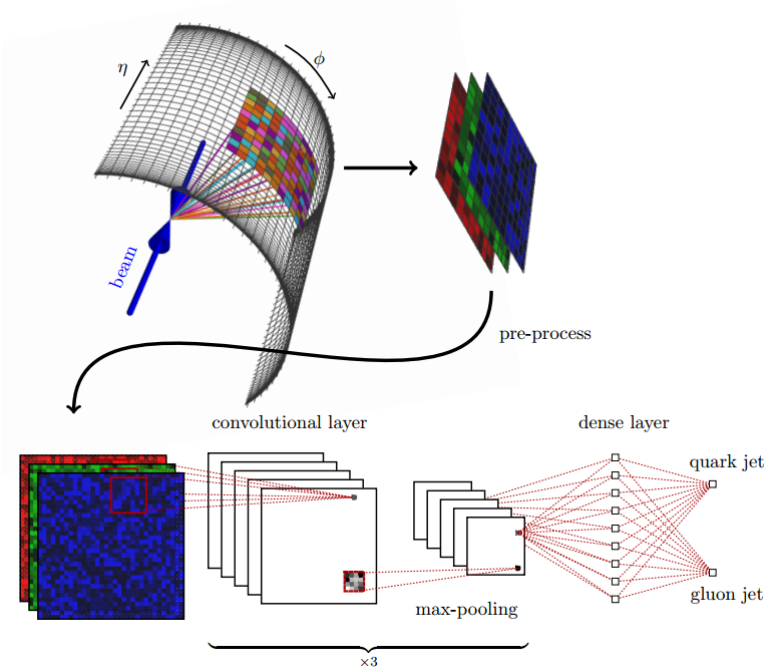
In particular, I will mainly talk about *jet physics* today.







Jet Classification
Quark-gluon tagging and more



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

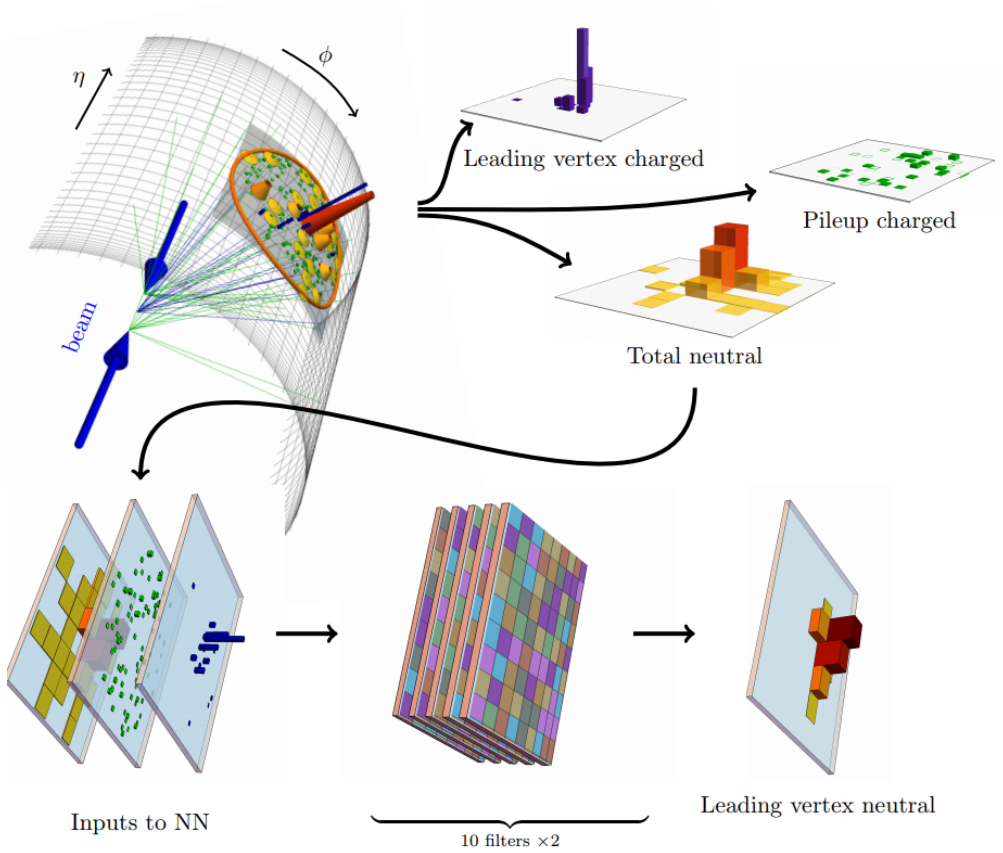
"Unroll"
cylindrical detector
(e.g. ATLAS, CMS)
to form *jet images*

- 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

$$\text{Theory} \to \text{Collision} \to \text{Decay (hadronization)}$$
$$\to \text{Detector simulation, electronics} \to \ldots \text{ and more}$$

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

Jet Simulation with GAN's

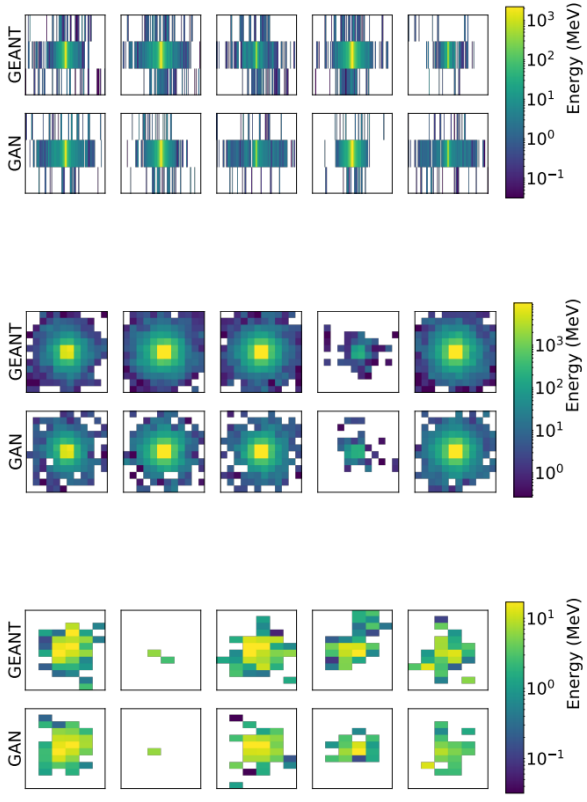
Monte Carlo simulation of jets @ LHC is *extremely* slow

$$\text{Theory} \to \text{Collision} \to \text{Decay (hadronization)}$$
$$\to \text{Detector simulation, electronics} \to \ldots \text{ and more}$$

... 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

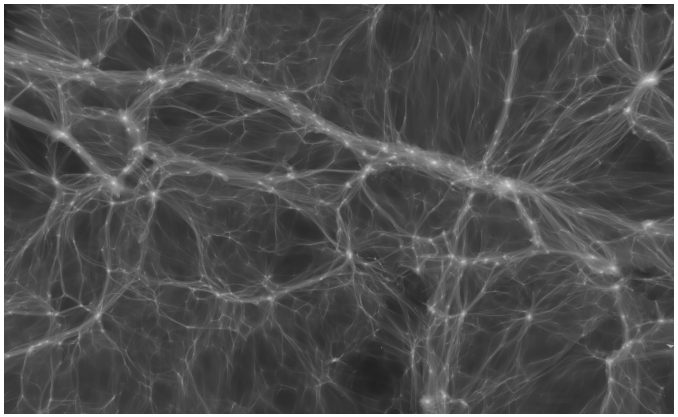


"CaloGAN" (2017)

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



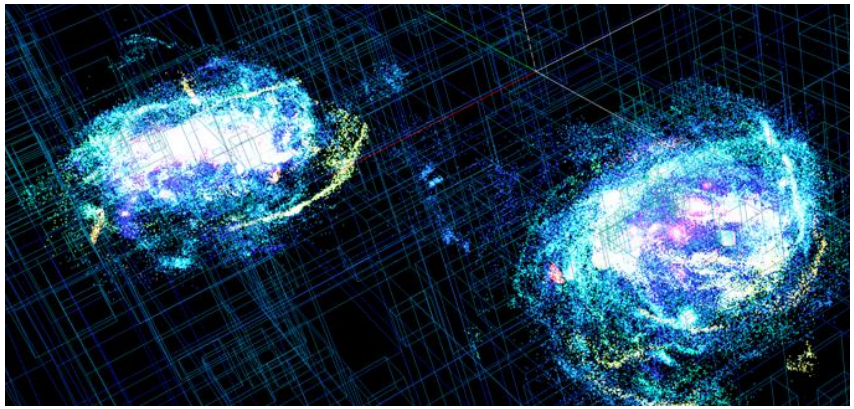


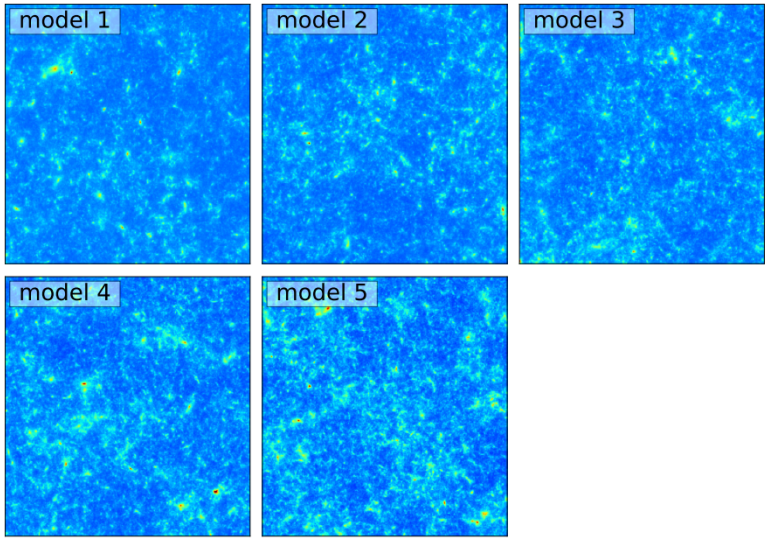
Image credit: Stanford/SLAC KIPAC
Computational Astrophysics & Dark Energy Groups

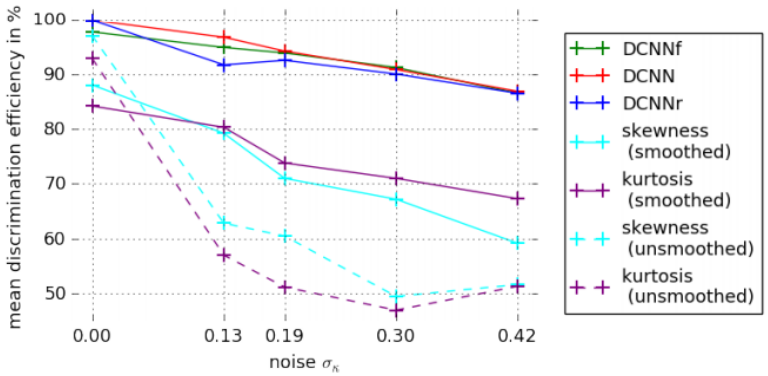
Cosmological Parameter Estimation
Classifying different, degenerate models

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*

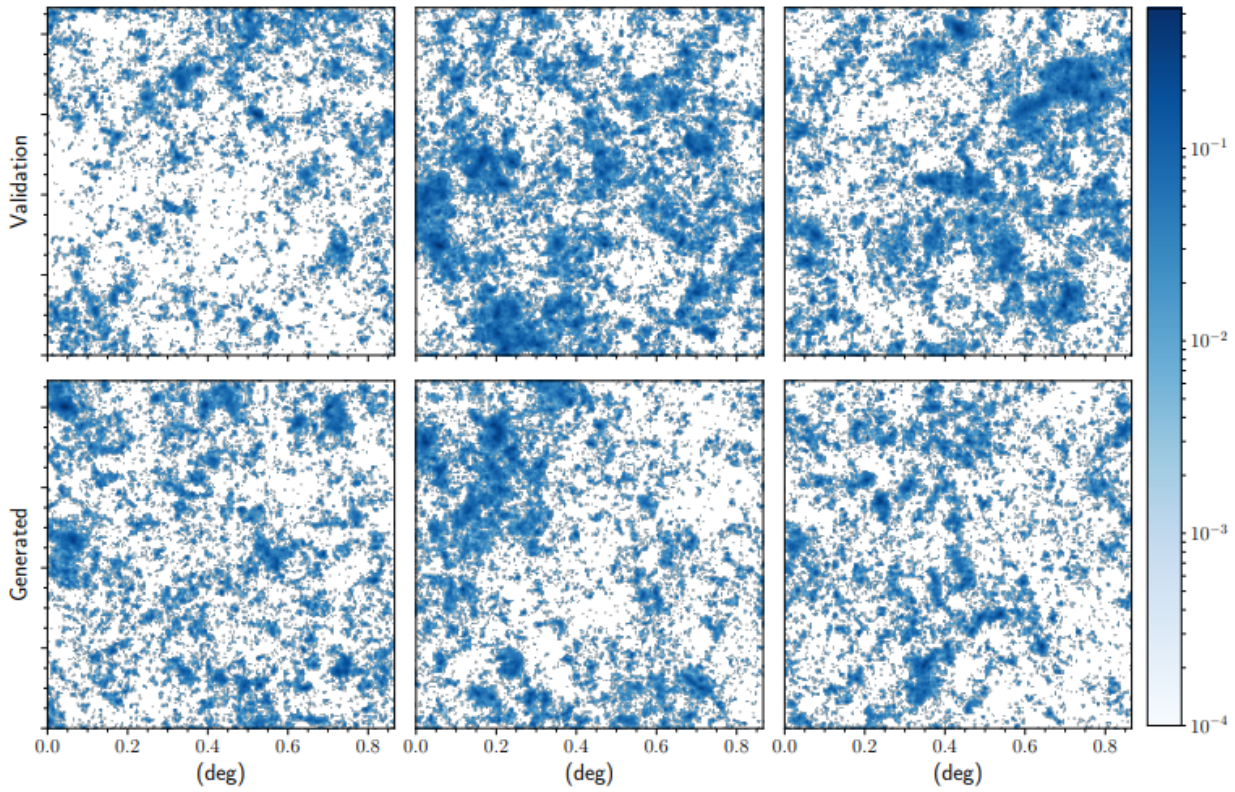




"Cosmological Model Discrimination with Deep Learning" (2017)

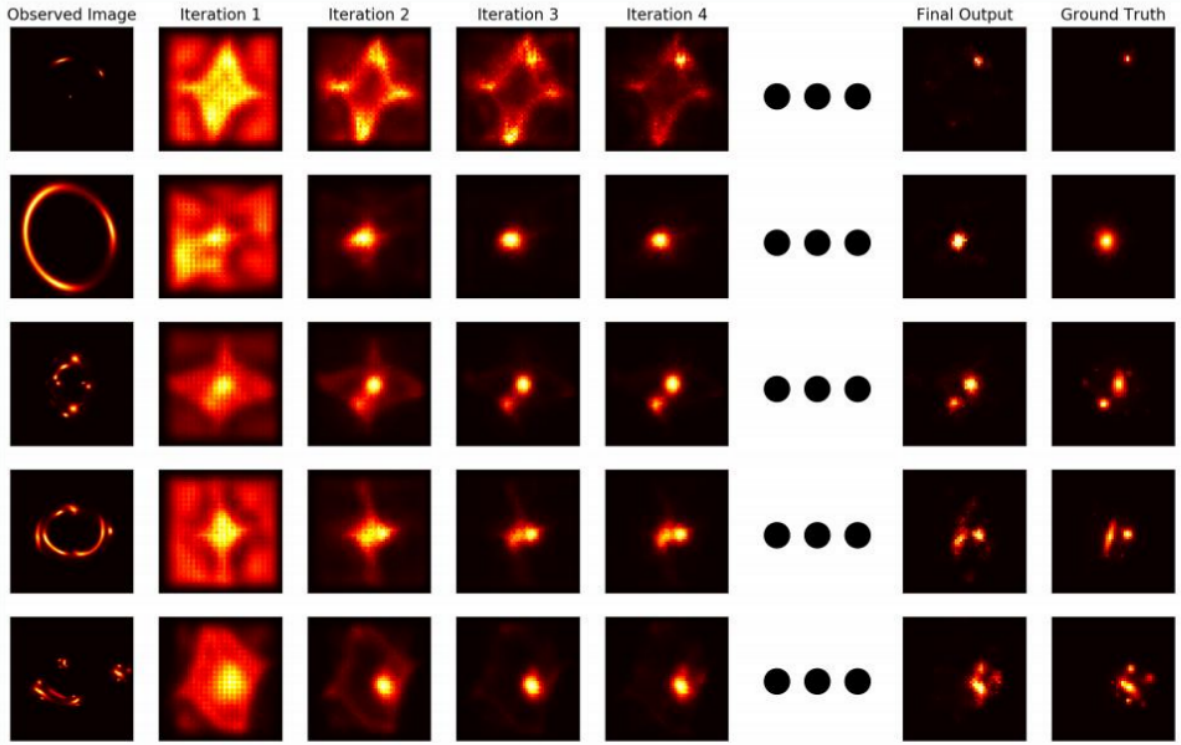
Cosmic-scale general-relativistic fluid dynamics simulation

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



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

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:

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

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"

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 :)

Reference

[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.

Reference (cont.)

[3] Sho Sonoda and Noboru Murata.

Neural network with unbounded activation functions is universal approximator.

Applied and Computational Harmonic Analysis, 43(2):233 – 268, 2017.

[4] The Apache Software Foundation.

Incubator mxnet: Image classification example.

`https://github.com/apache/incubator-mxnet/tree/master/example/image-classification`, 2018.

Reference (cont.)

[5] 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.

[6] 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.

Reference (cont.)

[7] 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.

[8] 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.

Reference (cont.)

[9] Kevin Frans.

Outline colorization through tandem adversarial networks.

CoRR, abs/1704.08834, 2017.

[10] Kamyar Nazeri and Eric Ng.

Image colorization with generative adversarial networks.

CoRR, abs/1803.05400, 2018.

[11] 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.

Reference (cont.)

[12] Jiale Zhi.

Pixelbrush: Art generation from text with gans.

`http://cs231n.stanford.edu/reports/2017/pdfs/322.pdf`, 2017.

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

Neural style transfer: A review.

CoRR, abs/1705.04058, 2017.

Reference (cont.)

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

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

JHEP, 01:110, 2017.

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

Pileup mitigation with machine learning (pumml).

2017, 07 2017.

Reference (cont.)

[16] 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.

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

Cosmological model discrimination with Deep Learning.

2017.

Reference (cont.)

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

Creating virtual universes using generative adversarial networks.

06 2017.

[19] Warren Morningstar and Dylan Rueter.

Magnet: Gravitational lens source reconstruction using deep learning.

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