THE AUTOMATED PHYSICIST

PARTICLE PHYSICS IN THE ERA OF AI

Harrison B. Prosper
Florida State University

Colloquium
University of Florida, October 27, 2022

Outline

- A Brief History of AI
- Of Black Boxes and Stochastic Parrots
- The Automated Physicist
- The Near Future

A BRIEF HISTORY OF AI

Moveable type (Gutenberg Bible, 1456)

By NYC Wanderer (Kevin Eng) - originally posted to Flickr as Gutenberg Bible

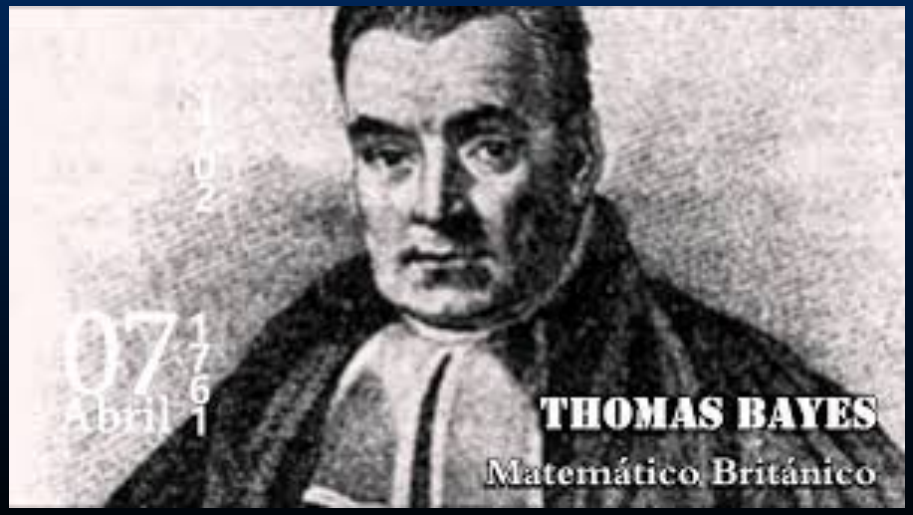
17th century

- Many philosophical ideas about knowledge, reason, and the nature of Man.

18th century

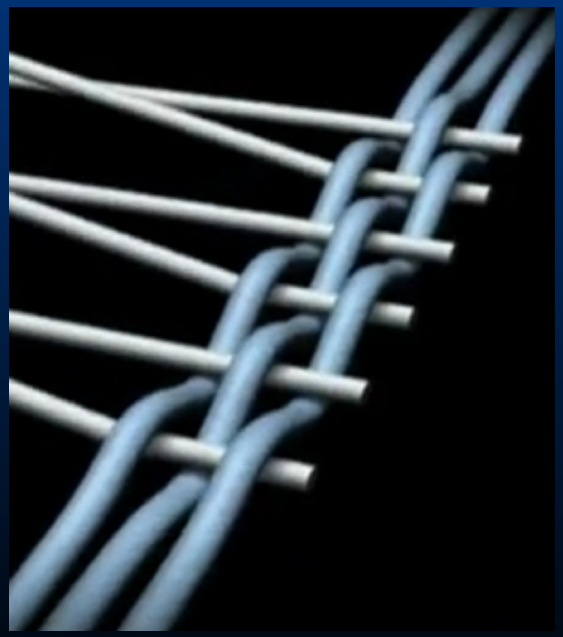
- 1763 – Thomas Bayes publishes important theorem.

$$P(\boldsymbol{H}|D) = \frac{P(D|\boldsymbol{H})P(\boldsymbol{H})}{P(D)}$$

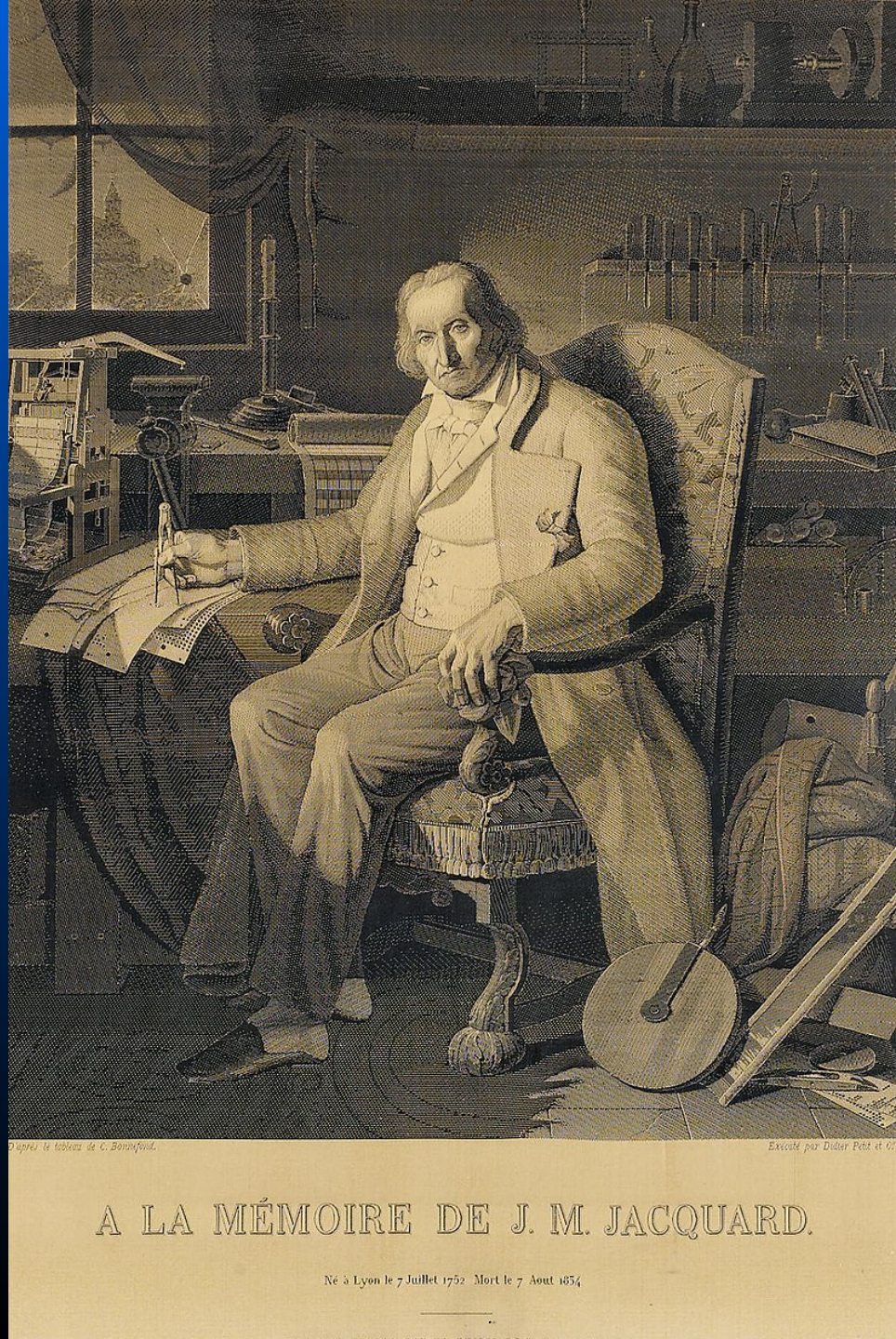


19th century

- 1801 – Joseph-Marie Jacquard invents first programmable machine.

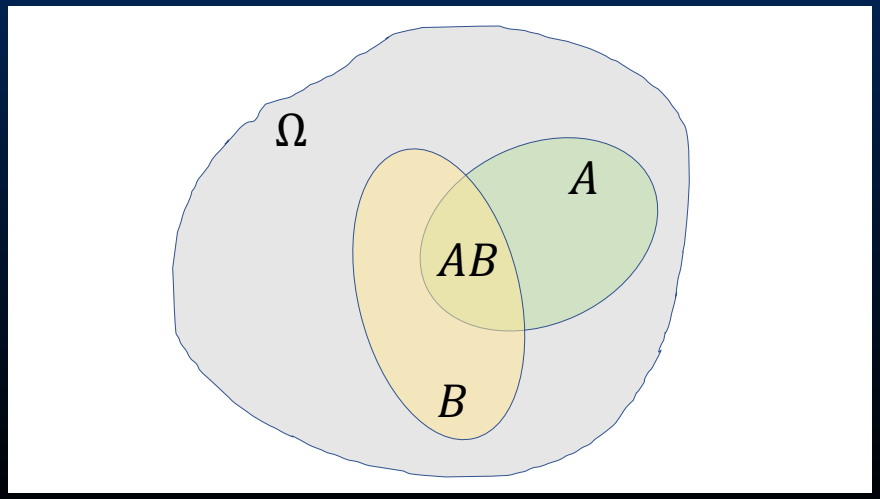


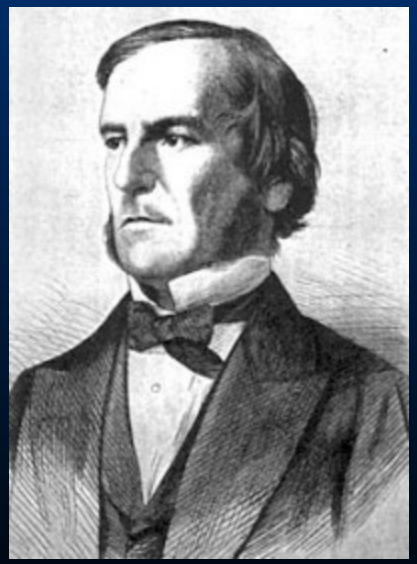
Wikimedia commons



19th century

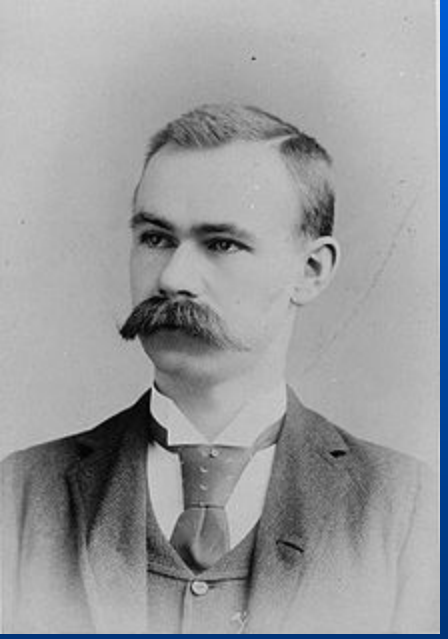
- 1832 – Charles Babbage designs first programmable calculator.
- 1854 – George Boole invents algebra of logic.





1815 - 1864

1890 US Census



Herman Hollerith
(1860 – 1929)

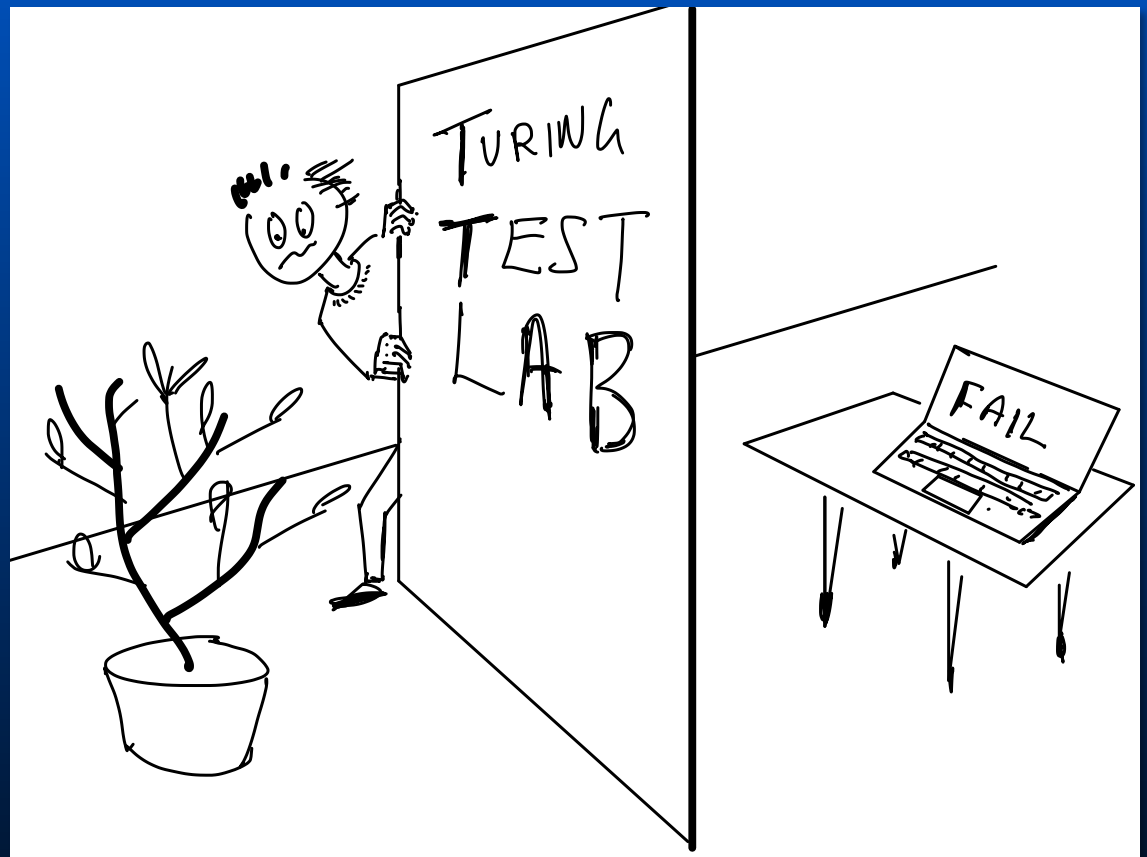
Wikimedia commons

https://www.census.gov/history/www/census_then_now/notable_alumni/herman_hollerith.html

1st Half of 20th Century

20th century (1900 – 1950)

- 1936 – Alan Turing proposes a universal computing machine.
- 1943 – Warren McCulloch and Walter Pitts invent *neural networks* (NN).
- 1950 – Turing Test, an operational definition of an artificially intelligent agent.



"George rethinks his life after failing the Turing Test"

HBP

2nd Half of 20th Century

20th century (1950 – 2000)

- Many important developments:

1. First industrial robot (George Devol's Unimate).
2. Development of specialized computer languages.
3. First robot able visually to locate and assemble objects (Edinburgh University).
4. Werbos invents *backpropagation algorithm.*
5. First autonomous robot rover on Mars (Sojourner, NASA, July 1997).

1997 World chess champion Gary Kasparov defeated by IBM's Deep Blue

Stan Honda/AFP/Getty Images

Computer Wins on ‘Jeopardy!’: Trivial, It’s Not

New York Times, Feb. 17, 2011

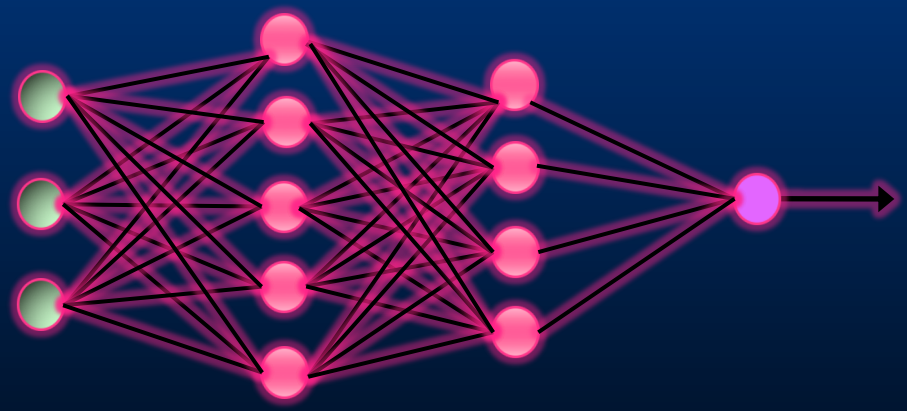
Carol Kaelson/Jeopardy Productions Inc., via Associated Press

Ken Jennings: “I felt obsolete”

TED Talk

The Deep Awakening

In 2006, University of Toronto researchers Hinton, Osindero, and Teh* developed a sophisticated practical method to train deep neural networks.

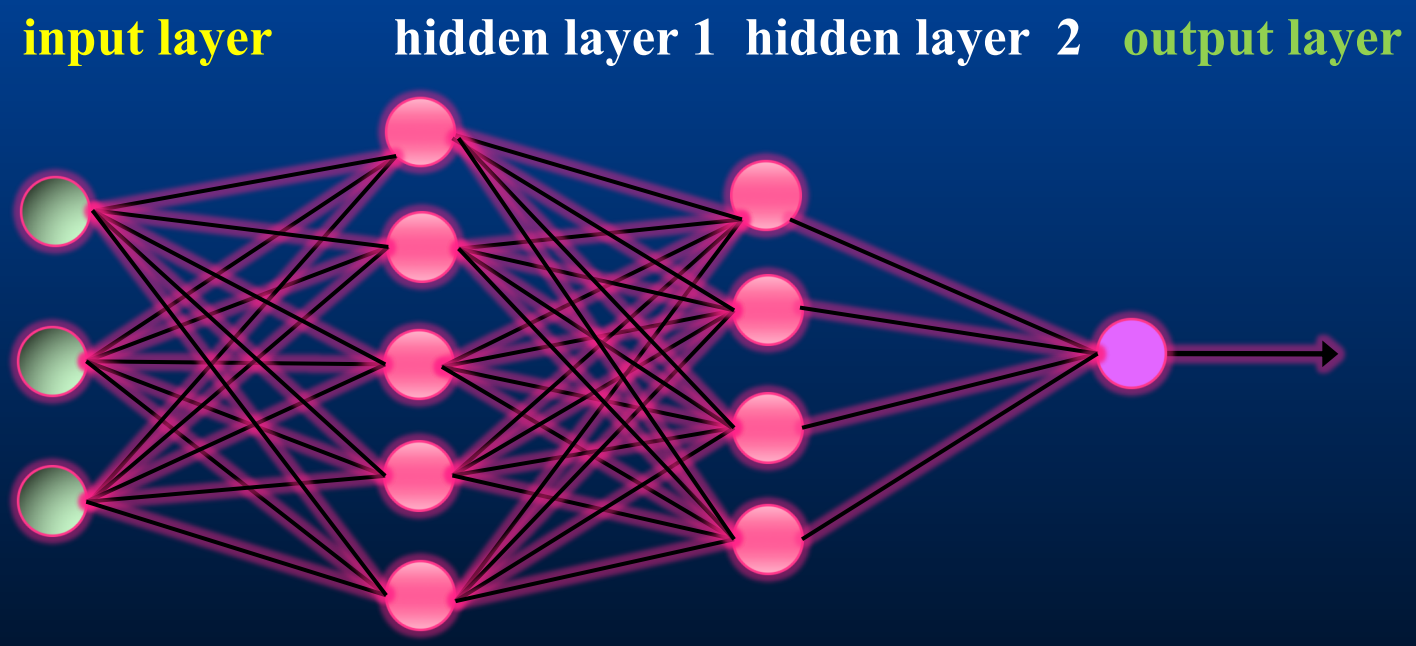


Geoffrey Hinton

* Hinton, G. E., Osindero, S. and Teh, Y., A fast learning algorithm for deep belief nets, Neural Computation 18, 1527-1554.

Deep Neural Networks

A deep neural network (DNN) with two “hidden” layers.



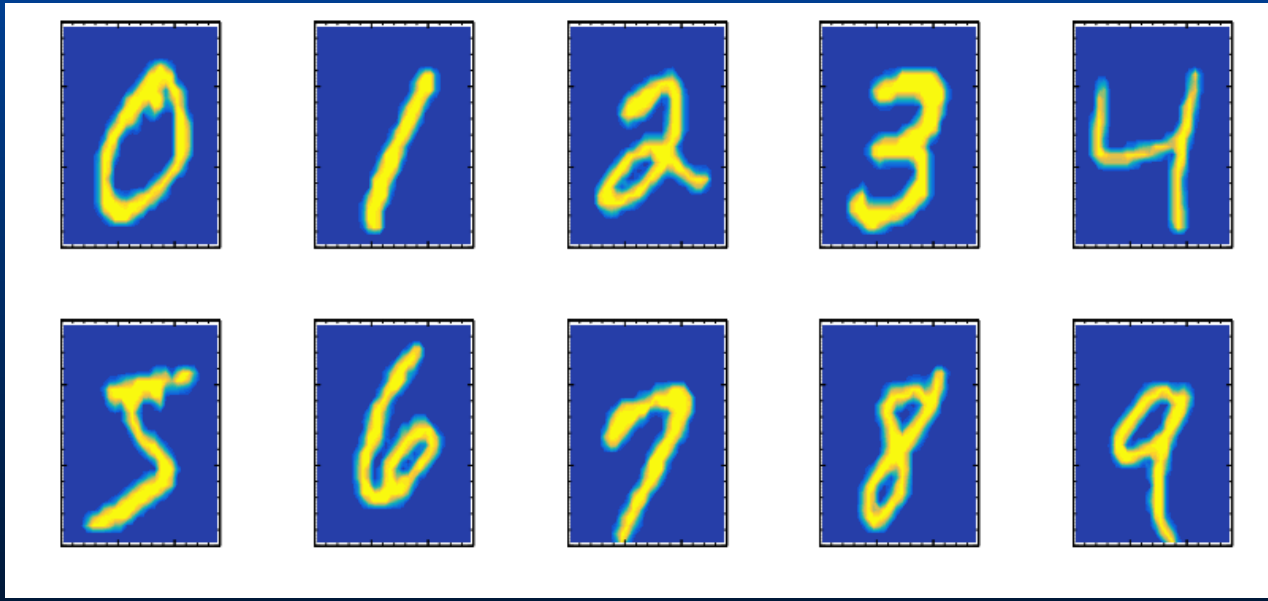
$\boldsymbol{x}$

$\boldsymbol{h}_1(\mathbf{c}_1 + \mathbf{d}_1\boldsymbol{x})$

$\boldsymbol{h}_2(\mathbf{c}_2 + \mathbf{d}_2\boldsymbol{h}_1)$

$\mathbf{o} = \boldsymbol{h}_3(\mathbf{c}_3 + \mathbf{d}_3\boldsymbol{h}_2)$

But, perhaps, sophistication may be overrated!*

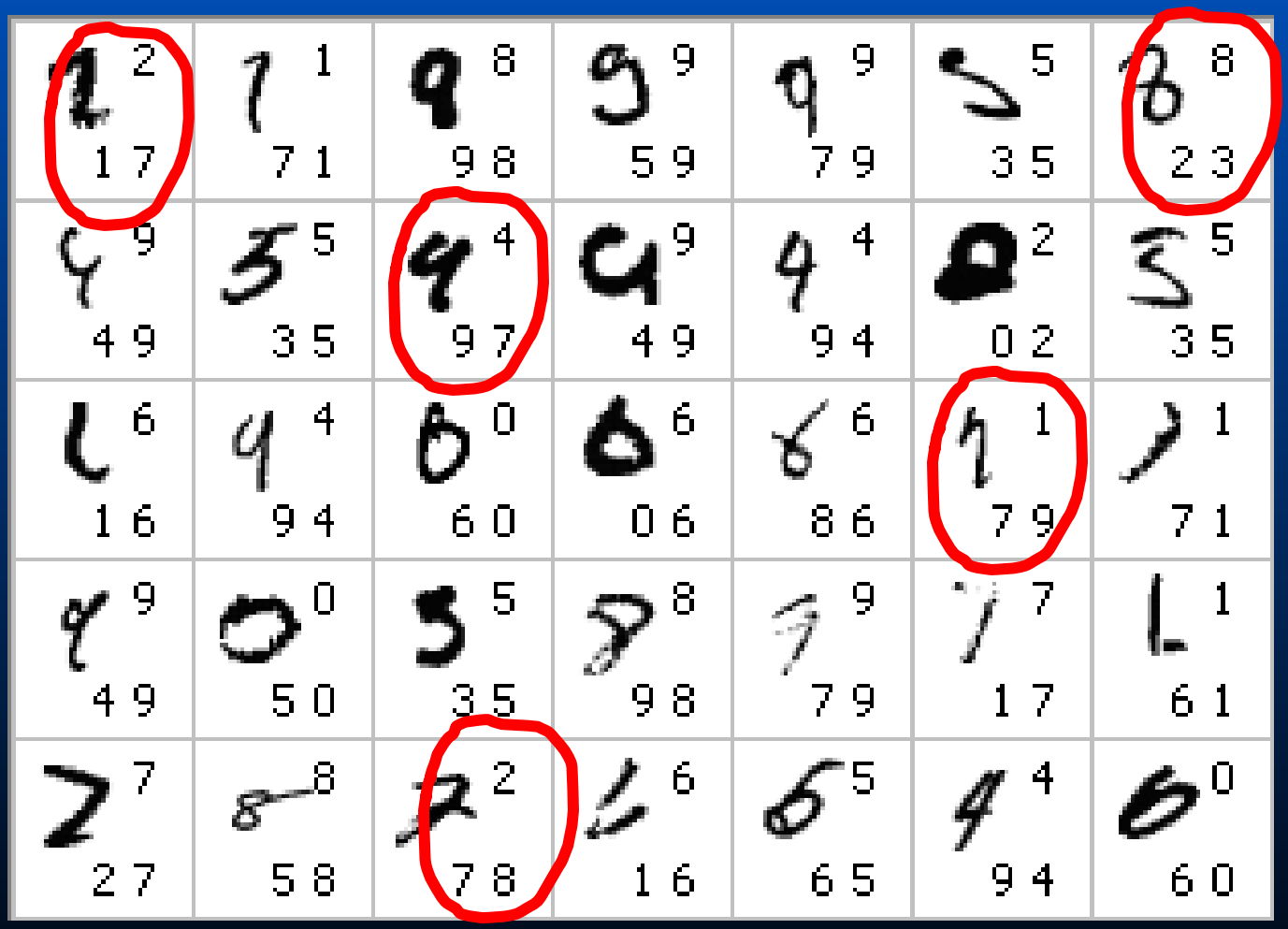


*Cireşan DC, Meier U, Gambardella LM, Schmidhuber J. ,

Deep, big, simple neural nets for handwritten digit recognition.

Neural Comput. 2010 Dec. 22 (12): 3207-20.

(784, 2500, 2000, 1500, 1000, 500, 10)



Upper right: correct answer; lower left answer of highest DNN output; lower right answer of next highest DNN output.

ARTICLE

doi:10.1038/nature24270

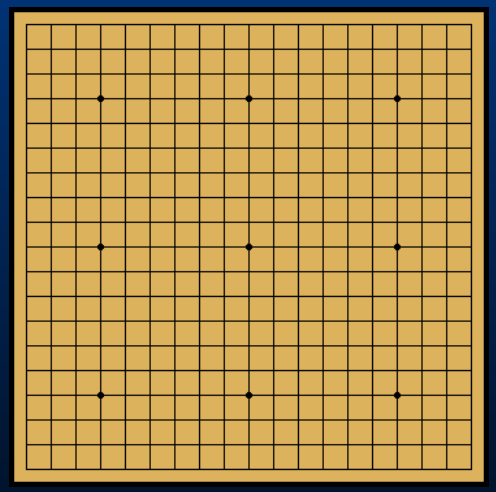
Mastering the game of Go without human knowledge

David Silver[1]*, Julian Schrittwieser[1]*, Karen Simonyan[1]*, Ioannis Antonoglou[1], Aja Huang[1], Arthur Guez[1], Thomas Hubert[1], Lucas Baker[1], Matthew Lai[1], Adrian Bolton[1], Yutian Chen[1], Timothy Lillicrap[1], Fan Hui[1], Laurent Sifre[1], George van den Driessche[1], Thore Graepel[1] & Demis Hassabis[1]

A long-standing goal of artificial intelligence is an algorithm that learns, *tabula rasa*, superhuman proficiency in challenging domains. Recently, AlphaGo became the first program to defeat a world champion in the game of Go. The tree search in AlphaGo evaluated positions and selected moves using deep neural networks. These neural networks were trained by supervised learning from human expert moves, and by reinforcement learning from self-play. Here we introduce an algorithm based solely on reinforcement learning, without human data, guidance or domain knowledge beyond game rules. AlphaGo becomes its own teacher: a neural network is trained to predict AlphaGo's own move selections and also the winner of AlphaGo's games. This neural network improves the strength of the tree search, resulting in higher quality move selection and stronger self-play in the next iteration. Starting *tabula rasa*, our new program AlphaGo Zero achieved superhuman performance, winning 100–0 against the previously published, champion-defeating AlphaGo.

Machine 4, Human 1

2016 – Google's DeepMind AlphaGo program beats Go champion Lee Sodol.



Photograph: Yonhap/Reuters

AlphaGo Zero

.AI] 5 Dec 2017

Mastering Chess and Shogi by Self-Play with a General Reinforcement Learning Algorithm

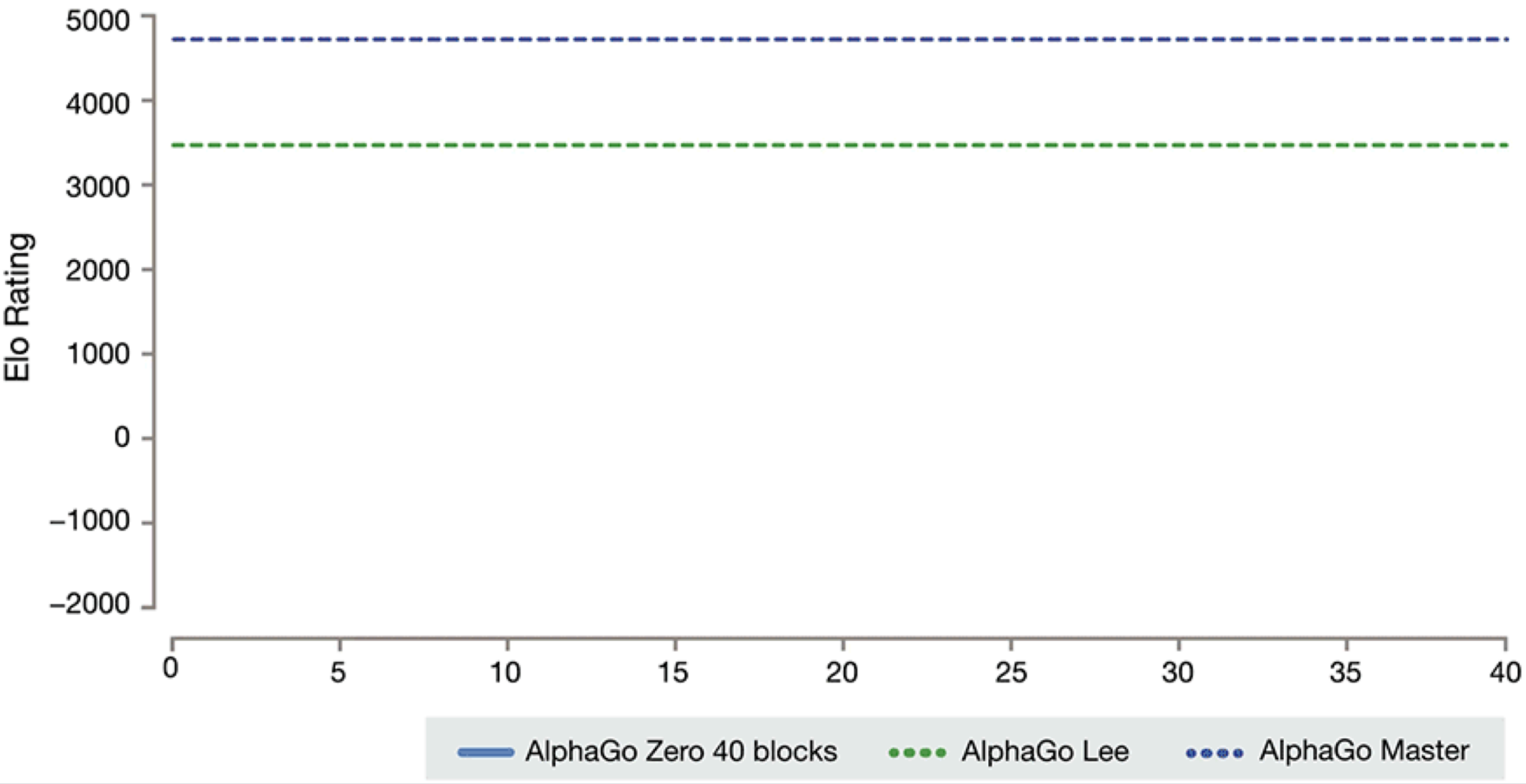
David Silver,[1*] Thomas Hubert,[1*] Julian Schrittwieser,[1*]
Ioannis Antonoglou,[1] Matthew Lai,[1] Arthur Guez,[1] Marc Lanctot,[1]
Laurent Sifre,[1] Dharshan Kumaran,[1] Thore Graepel,[1]
Timothy Lillicrap,[1] Karen Simonyan,[1] Demis Hassabis[1]

[1]DeepMind, 6 Pancras Square, London N1C 4AG.
*These authors contributed equally to this work.

"Starting from random play, and given no domain knowledge except the game rules, *AlphaGo Zero* achieved within 24 hours a superhuman level of play in the games of chess and shogi (Japanese chess) as well as Go, and convincingly defeated a world-champion program in each case."

arXiv:1712.01815v1

https://deepmind.com/blog/alphago-zero-learning-scratch/



…the not-too-distant future

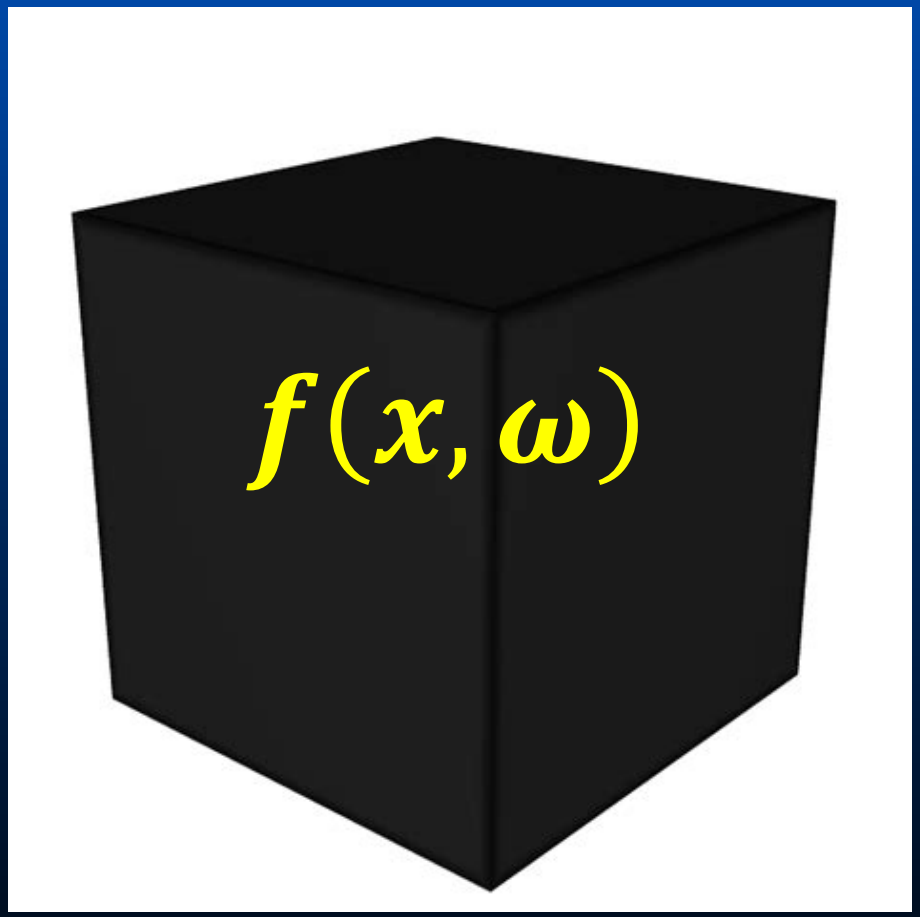
"University of Florida professors protest their replacement by iPhone 9000s"

New York Times, Feb. 8, 2076

OF BLACK BOXES AND STOCHASTIC PARROTS

Of Black Boxes

$x \rightarrow$



$\rightarrow y$

Of Black Boxes

Almost all machine learning models $f(x, \omega)$ are "trained", that is, fitted to data $D = \{(x_i, y_i)\}$ by minimizing an empirical risk function,

$$R_N = \frac{1}{N} \sum_i L(y_i, f_i), \qquad f_i = f(x_i, \omega)$$

where $L(y_i, f_i)$ is a loss function and ω are the parameters of the model.

Now let $N \to \infty$

Of Black Boxes

In the limit $N \to \infty$, the empirical risk function becomes the risk functional

$$R[f] = \int \int L(y, f)\, p(x, y)\, dxdy$$

where $p(x, y)$ is the probability distribution of the training data.

Of Black Boxes

When $R[f]$ is minimized with respect to the model $f(x, \omega)$, one finds

$$\boxed{\int \frac{\partial L}{\partial f}\, p(y \mid x)\, dy = 0}$$

This equation suggests that fretting about black boxes misses the point, especially if you're a scientist!

But fretting about the training data is warranted if you're data comes from web crawling!

Of Black Boxes

Example 1 (quadratic loss):

$$L = (y - f)^2$$

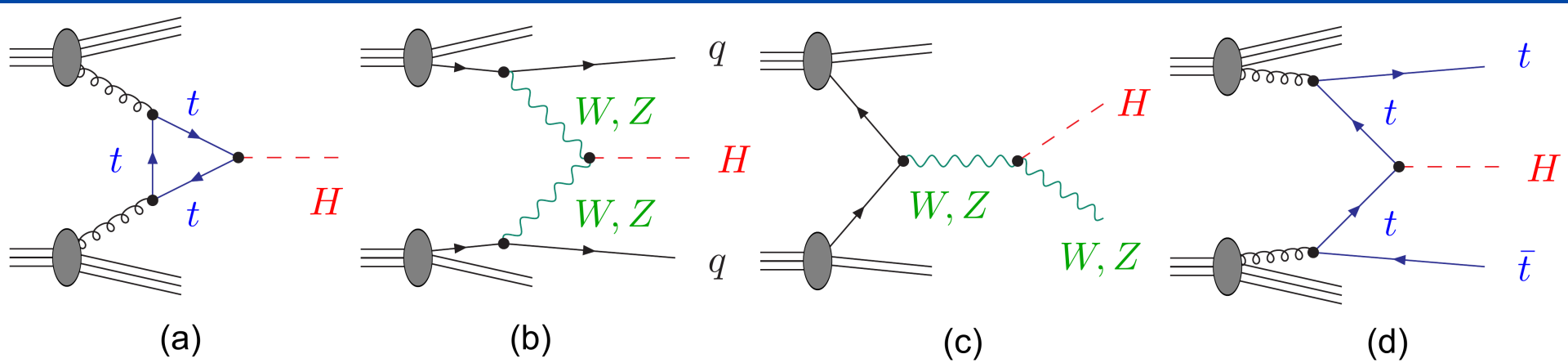
$$\Longrightarrow f \approx \int y\, p(y \mid x)\, dy$$

Example 2 (cross-entropy loss):

$$L = -[y \log f + (1 - y) \log(1 - f)], \quad y \in \{0, 1\}$$

$$\Longrightarrow f \approx p(y = 1|x)$$

Eg.: Higgs Boson Production @ LHC



Process		$\sigma \times BR$ (fb)
(a) Gluon gluon fusion	(ggF)	12.18
(b) Vector boson fusion	(VBF)	1.044
(c) Associated production	(VH)	1.047
(d) Top anti-top fusion	(ttH)	0.393

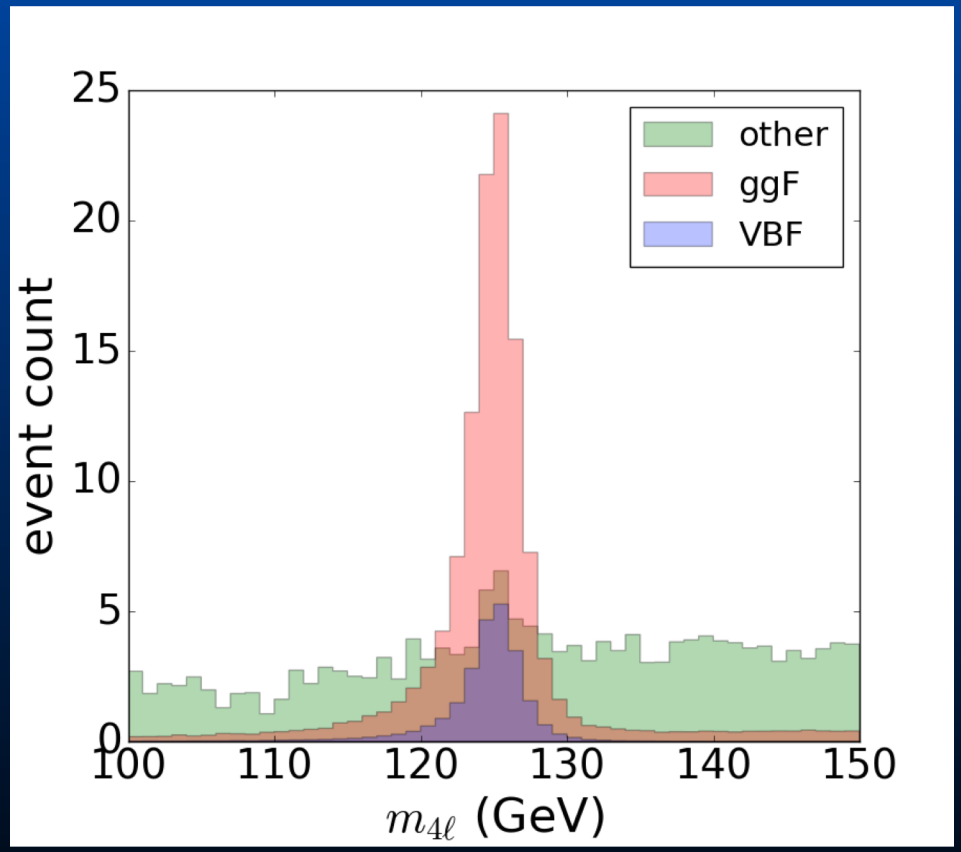
Before event selection, background ~ 1700 times larger!

http://www.scholarpedia.org/article/The_Higgs_Boson_discovery

Eg.: Higgs Boson Production @ LHC

The Higgs boson mass is an excellent discriminant between Higgs boson events and many other classes of events at the Large Hadron Collider.

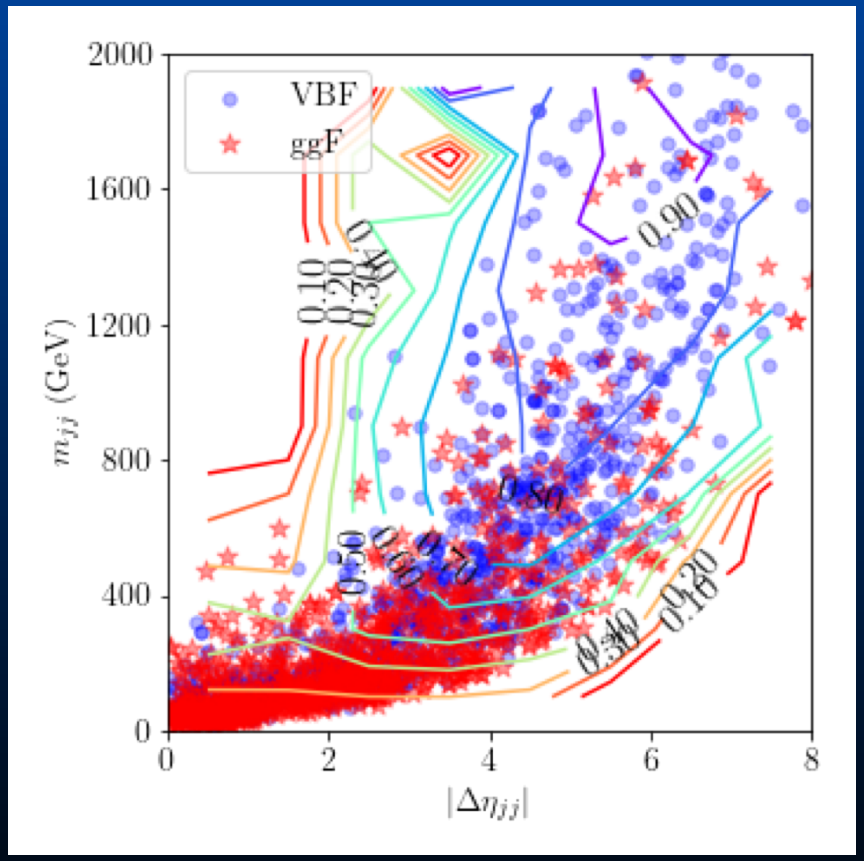
But obviously we need other observables to separate VBF events from ggF events.



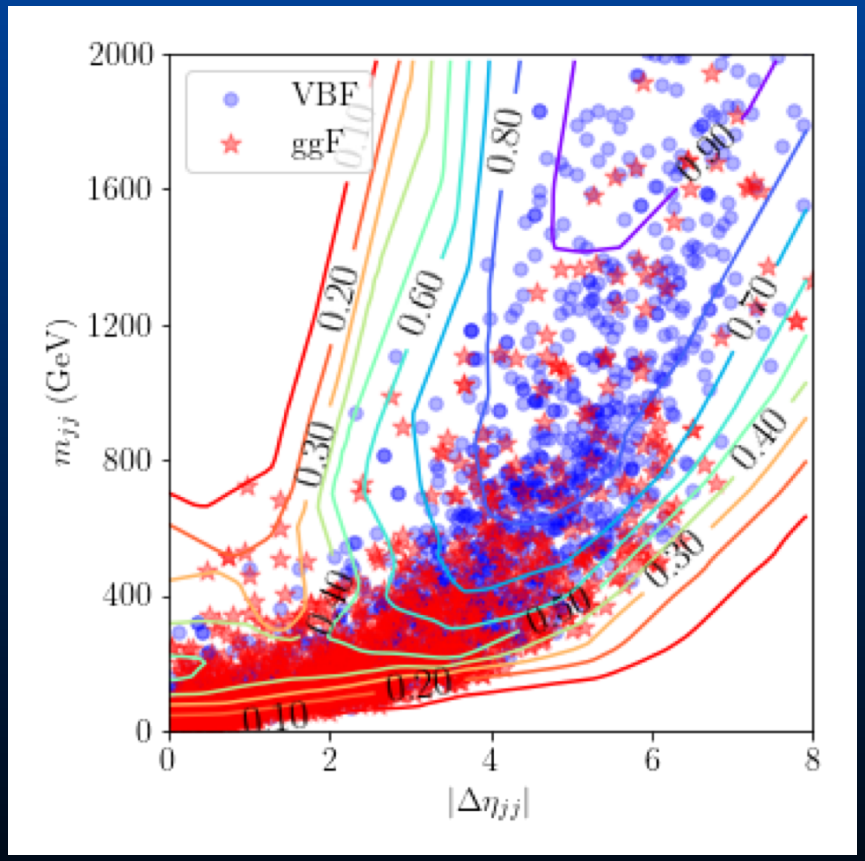
Eg.: Higgs Boson Production @ LHC

$$\frac{VBF}{VBF + ggF} \approx p(y = 1|x)$$

$$f(x, \omega) \approx p(y = 1|x)$$



Using histograms of VBF and ggf

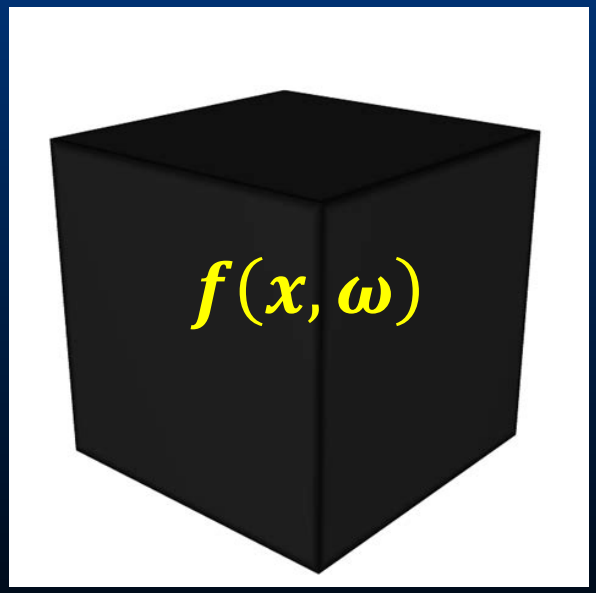


Using a simple ML model

…and Stochastic Parrots

Even the most advanced language model is still a statistical machine…

My name is Elvis →

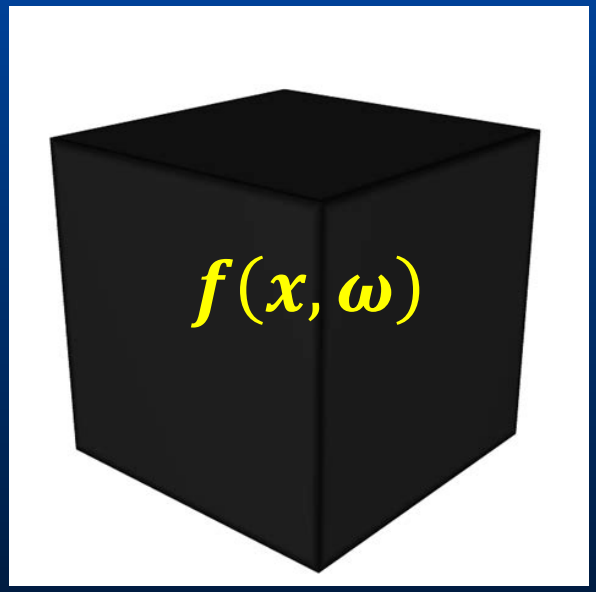


→ Je m'appelle Elvis

Emily Bender, https://faculty.washington.edu/ebender/stochasticparrots.html

Of Black Boxes And Stochastic Parrots

$x \rightarrow$



$\rightarrow y$

Garbage **I**n **G**arbage **O**ut!

THE AUTOMATED PHYSICIST

“That is positively the dopiest idea I have heard.”

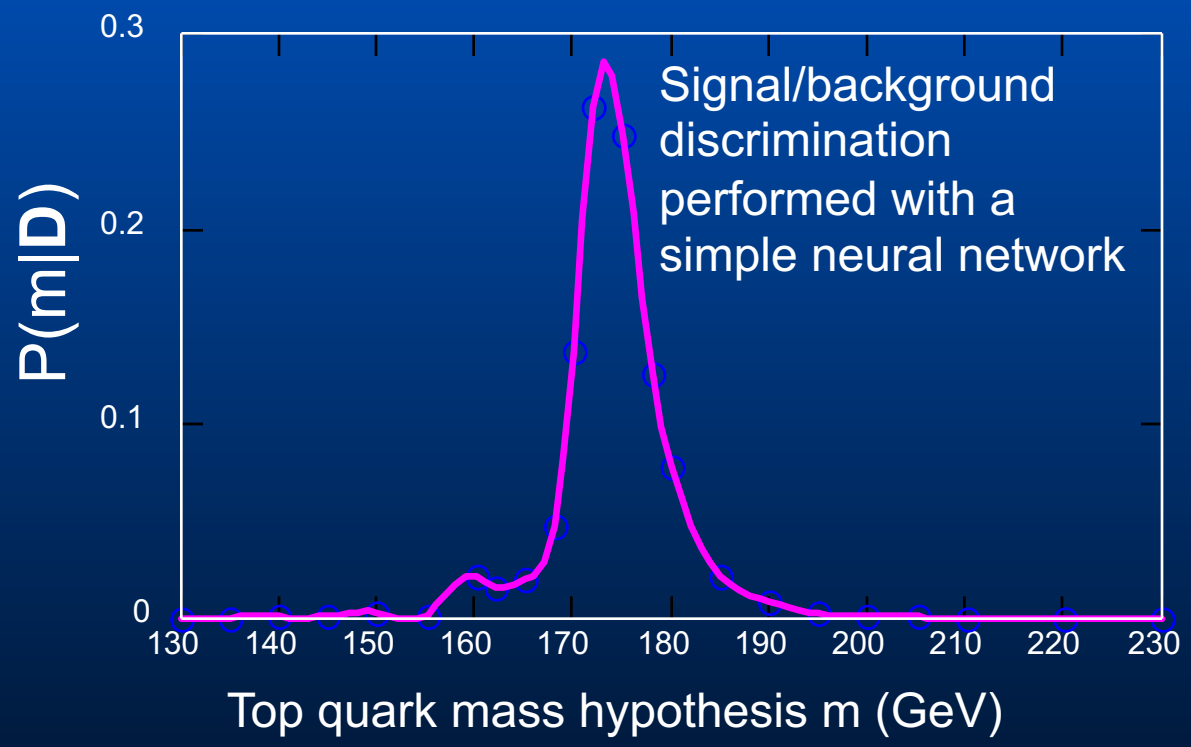
Richard Feynman,

Thinking Machines Corporation, summer 1983

Machine Learning in HEP, The Early Days

- 1988 Denby, Comp. Phys. Comm.49:429 (1988)
- 1990 Bhat, Lönnblad, Meier, Sugano, *Snowmass;* Lönnblad, Peterson, Rögnvaldsson, *Phys. Rev. Lett*. 65:1321 (1990)
- 1992 Peterson, CHEP 92, Denby, FERMILAB-CONF-92-269-E (1992)
- 1994 Bhat PC (for the DØ Collaboration), APS Meeting, Albuquerque, NM
- 1997 Moneti (CLEO Collaboration) Nuclear Physics B (Proc. Suppl.) 59:17 (1997)

Top Quark Mass (DØ, 1997)



mass = 173.5 ± 4.5 GeV (173.34 ± 0.76 GeV*)

signal = 33 ± 8 events

background = 50.8 ± 8.3 events

Pushpalatha Bhat, HBP

*arXiv:1403.4427

Symbolic Mathematics

In December 2019, Guillaume Lample and François Charton* at Facebook AI Research, Paris, made the startling claim: "*We achieve results that outperform commercial Computer Algebra Systems such as Matlab or Mathematica*."

Lample

Charton

G. Lample and F. Charton, Deep Learning for Symbolic Mathematics, arXiv: 1912.01412v1.

Symbolic Mathematics

The authors' system simplifies, integrates functions, and solves 1st and 2nd order differential equations.

The training data are pairs (x, t) of correctly formed, *randomly generated*, expressions x with associated solutions t.

For example, for integration, at least two approaches are used:

1. Forward: (x, t) where $t = \int x$
2. Backward: (x, t) where $x = Dt$

https://github.com/facebookresearch/fairseq

Symbolic Mathematics

…and here is something noteworthy…

The authors trained their model using the subset of randomly generated functions that sympy can integrate, e.g.,

```
import sympy as sm
z = sm.Symbol('z')
x = sm.exp(-z)*sm.cos(z)
t = sm.integrate(x, z)
x, t
```

$$\left(e^{-z}\cos(z),\ \frac{e^{-z}\sin(z)}{2} - \frac{e^{-z}\cos(z)}{2}\right)$$

…and found that the model was able to integrate functions that sympy could not!

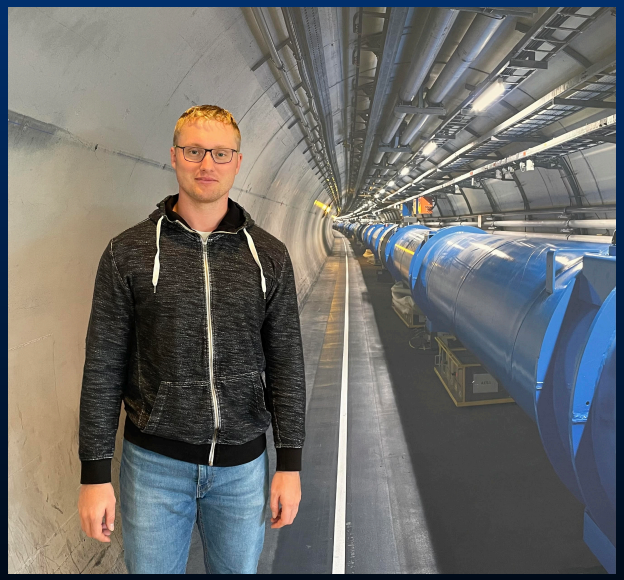
SYMBA

SYMBA: Symbolic Computation of Squared Amplitudes in High Energy Physics with Machine Learning

Abdulhakim Alnuqaydan

University of Kentucky

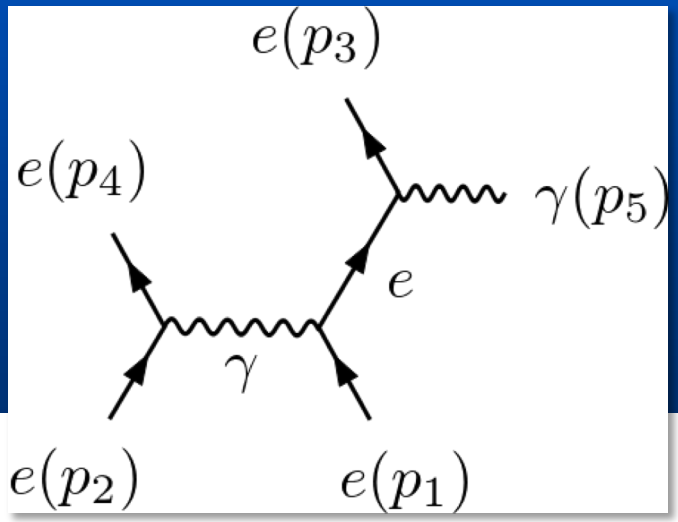
Marco Knipfer

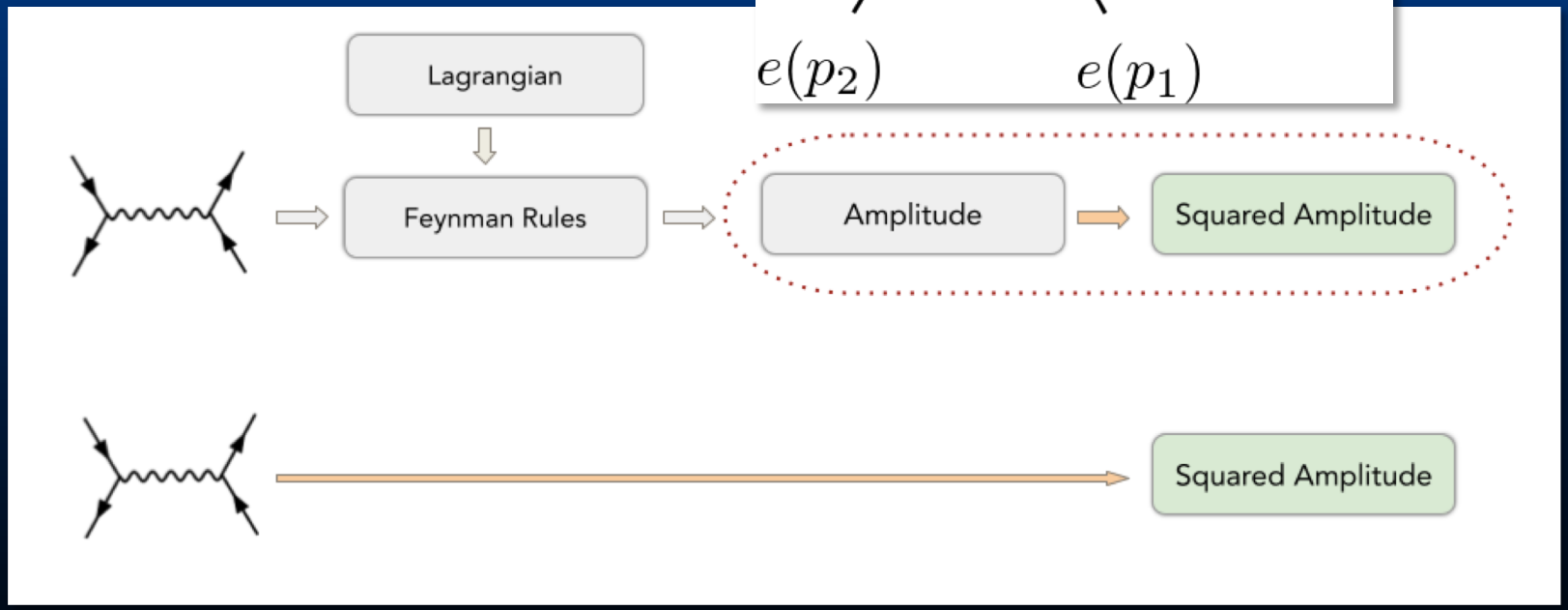


University of Alabama

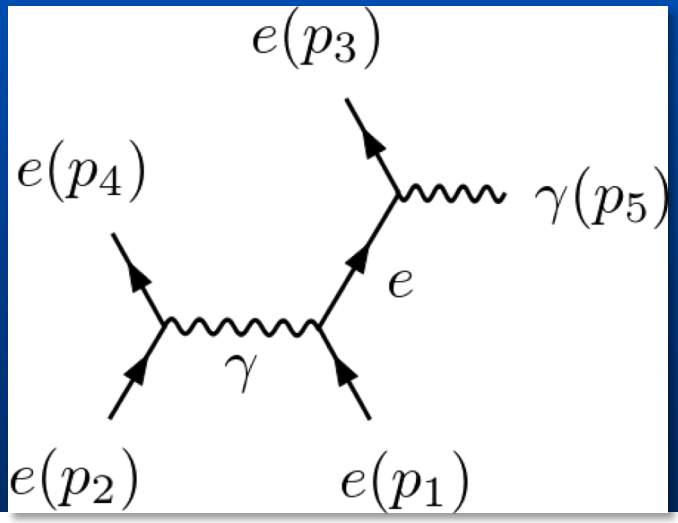
Prof. Sergei Gleyzer and HBP

SYMBA


$e(p_3)$
$e(p_4)$
$\gamma(p_5)$
e
γ
$e(p_2)$
$e(p_1)$


Lagrangian
Feynman Rules
Amplitude
Squared Amplitude
Squared Amplitude

SYMBA



- ***The amplitude*** $(e\ e \to e\ e\ \gamma)$:

$$i\mathcal{M} = \frac{\frac{1}{2}\, ie^3 \Big(p_{3\rho}\, \gamma^{\rho}_{\epsilon}\, \gamma_{\rho\eta}\, A^{\rho *}_{j}(p_5)\, \mathbf{e}^{*}_{i\eta}(p_4)\, \mathbf{e}^{*}_{l\epsilon}(p_3)\, \mathbf{e}_{k\delta}(p_2)\, \mathbf{e}_{i\delta}(p_1) \; - \; \frac{1}{2}\, p_{5\sigma}\, \gamma_{\rho\epsilon}\, \gamma^{\rho}_{\epsilon}\, \gamma_{\rho\eta}\, \gamma^{\sigma}_{\epsilon}\, A^{\rho *}_{j}(p_5) \mathbf{e}^{*}_{i\eta}(p_4)\, \mathbf{e}^{*}_{l\epsilon}(p_3)\, \mathbf{e}_{k\delta}(p_2)\, \mathbf{e}_{i\delta}(p_1) \Big)}{((m_e^2 - \vec{p_2}.\vec{p_4}) \; * \; \vec{p_3}.\vec{p_5})}$$

- ***The squared amplitude*** $(e\ e \to e\ e\ \gamma)$:

$$|\mathcal{M}|^2 = -\frac{e^6}{((\vec{p_3}.\vec{p_5})^2 * (m_e^2 - \vec{p_2}.\vec{p_4})^2)} (2m_e^6 + m_e^4 * (-\vec{p_1}.\vec{p_3} - \vec{p_1}.\vec{p_5} - \vec{p_2}.\vec{p_4} + 2\vec{p_3}.\vec{p_5}) + m_e^2 * (\vec{p_1}.\vec{p_2} * \vec{p_3}.\vec{p_4} +$$
$$\vec{p_1}.\vec{p_2} * \vec{p_4}.\vec{p_5} + \vec{p_1}.\vec{p_4} * \vec{p_2}.\vec{p_3} + \vec{p_1}.\vec{p_4} * \vec{p_2}.\vec{p_5} + \vec{p_1}.\vec{p_5} * \vec{p_3}.\vec{p_4} - \vec{p_2}.\vec{p_4} * \vec{p_3}.\vec{p_5}) - \vec{p_1}.\vec{p_2} * \vec{p_3}.\vec{p_5} * \vec{p_4}.\vec{p_5} - \vec{p_1}.\vec{p_4} * \vec{p_2}.\vec{p_5} * \vec{p_3}.\vec{p_5})$$

SYMBA

MARTY

- https://marty.in2p3.fr
- A **M**odern **AR**tificial **T**heoretical ph**Y**sicist
- Complicated installation, worked together with developer to fix some bugs in the installer

Table 1: Amplitude-squared amplitude Model results

	Training Size	Sequence Acc.	Token Score	RMSE
QED (sequence)	136K	98.6%	99.6%	2×10^{-3}
QCD (sequence)	113K	87.8%	92.3%	0.11
(QED + QCD) on QED	249K	99.4%	99.9%	2×10^{-5}
(QED + QCD) on QCD	249K	89.0%	93.0%	0.08
QED (diagram)	127K	78.4%	84.0%	0.2

THE NEAR FUTURE

Machine Learning in Particle Physics

Evolution of the use of machine learning in physics:

- **traditional:** classification & regression
- **emerging:** inference & generation

Kyle Cranmer, ACAT 2017


THANK YOU!

“There are, therefore, agents in nature able to make the particles of bodies stick together by very strong attractions. And it is the business of experimental philosophy to find them out”

Sir Isaac Newton