

AI/ML In Science & Society

G. Watts (UW/Seattle)



Who am I?

Gordon Watts
Professor of Physics
gwatts@uw.edu

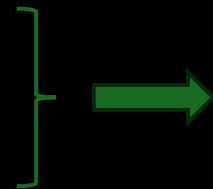
Joined UW in 1999 (!!)

Experimental Particle Physicist



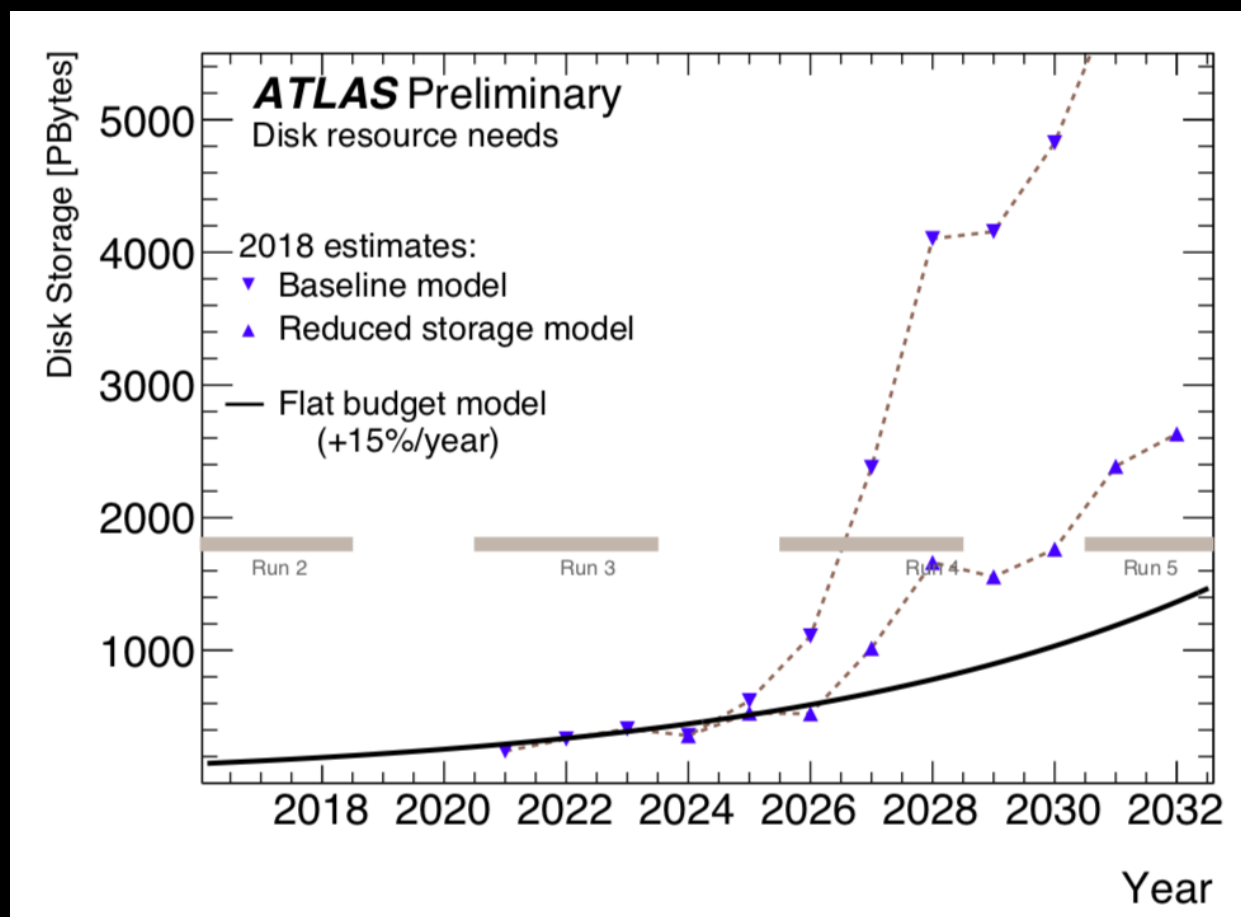
Interests:

- Intersection of physics, computing, and hardware
- Top Quark Physics
- Searches for Long Lived Data

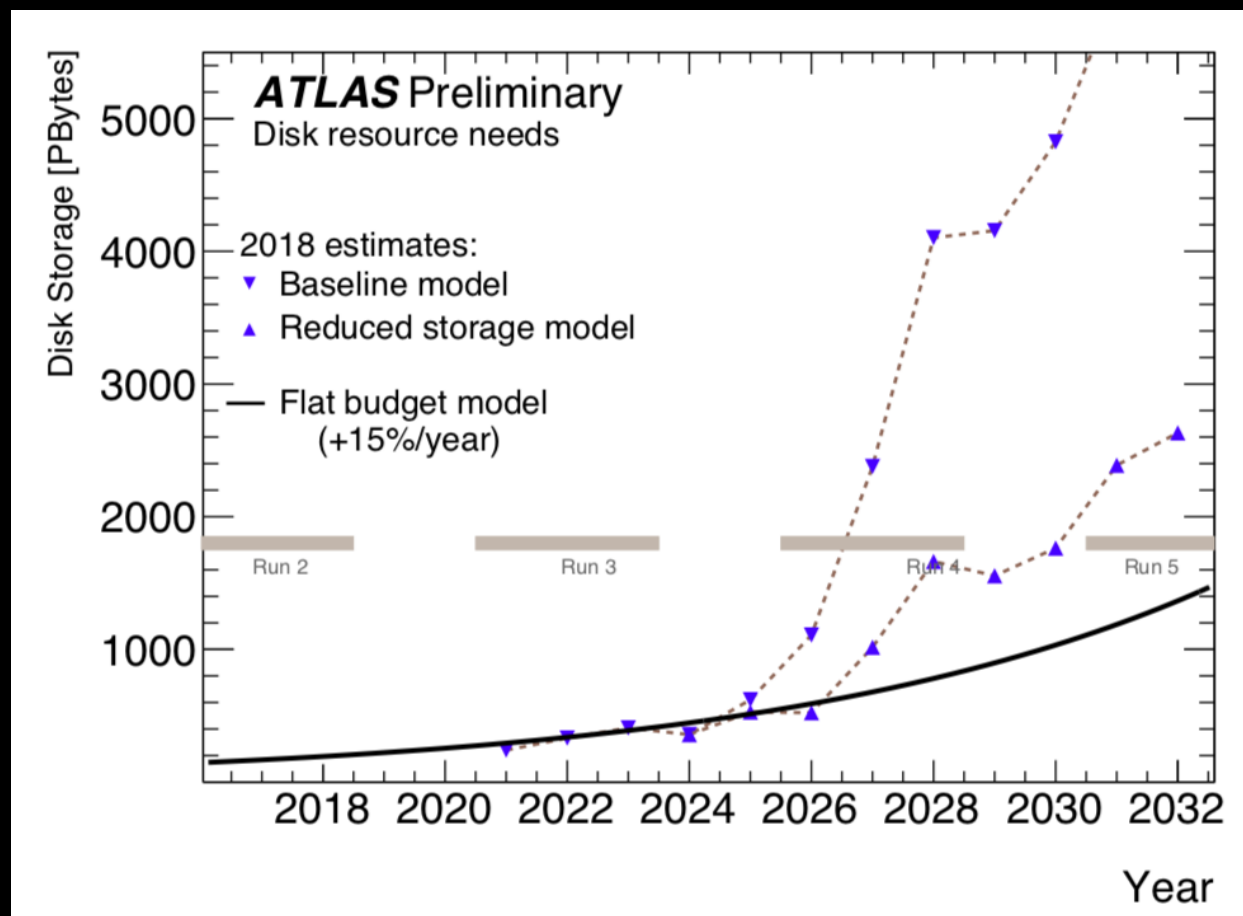


Machine Learning and AI

Why Do We Care About AI/ML In Science?



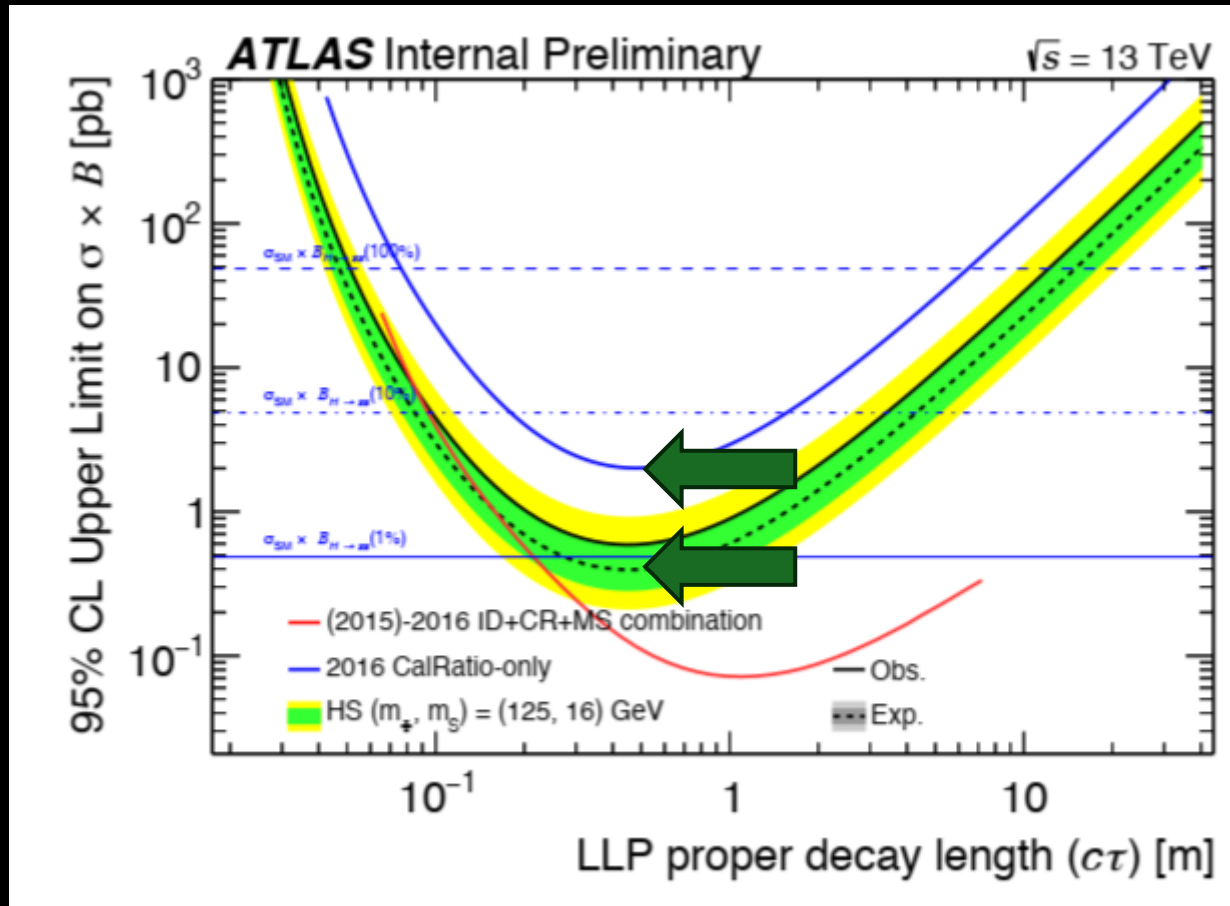
Why Do We Care About AI/ML In Science?



Big Data

Non-Trivial Pattern Recognition

Worth Money and Time!



This change in this plot is due to:

- Increase in data (about 50%)
- Use of more sophisticated Machine Learning technique (about 50%)

This translates into close to a year of running the Large Hadron Collider!

Electricity alone is ~90M USD

Everything else is another ~250M USD

My Personal Opinion

You should not get a Ph.D. without
knowing what ML is, where it is useful,
and how to use it.

(and likely how to code one up)

(how long till this is true for an undergraduate? Not long...)

Introduction To Deep Learning and Machine Learning

Artificial Intelligence

Human-like intelligent machines or programs

Machine Learning

AI algorithms that learn from data instead of being explicitly human programmed

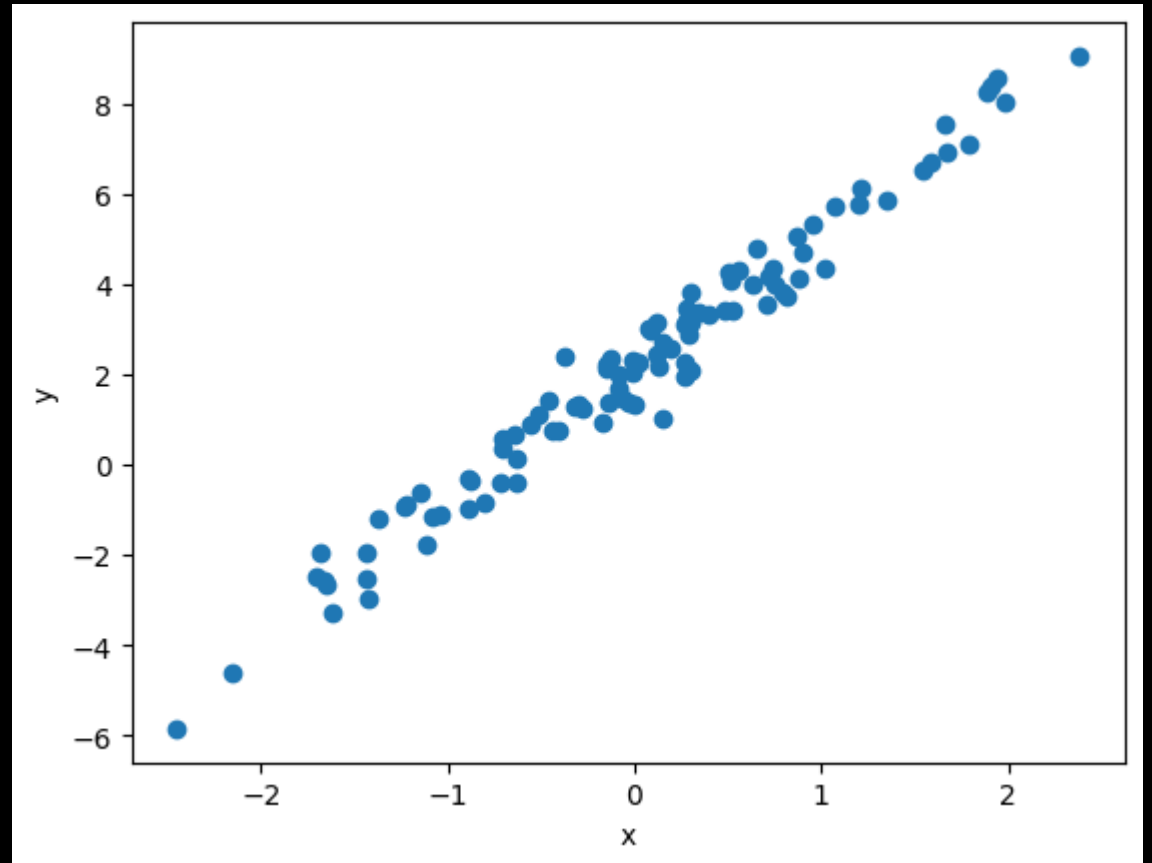
It is all about the Math!

But that is all it is about!

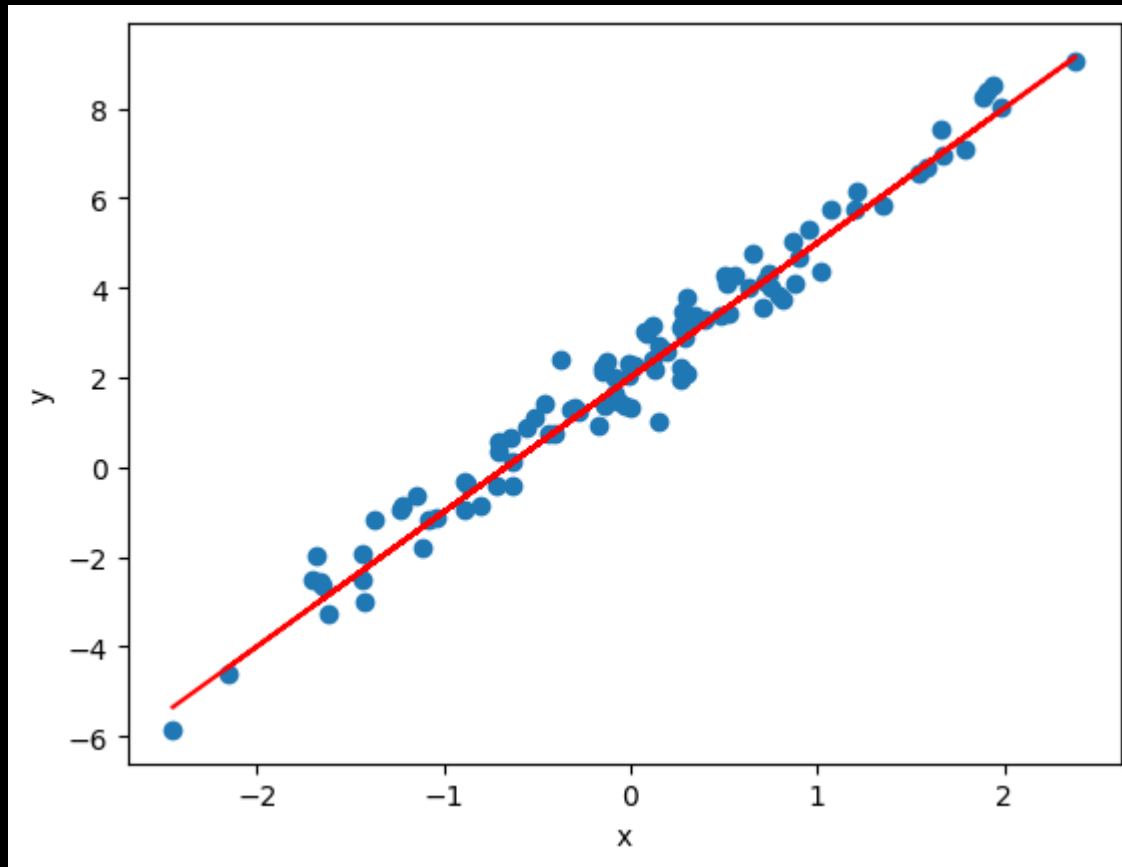
What is this?

Guess where the $x = 1.9$ value would be!

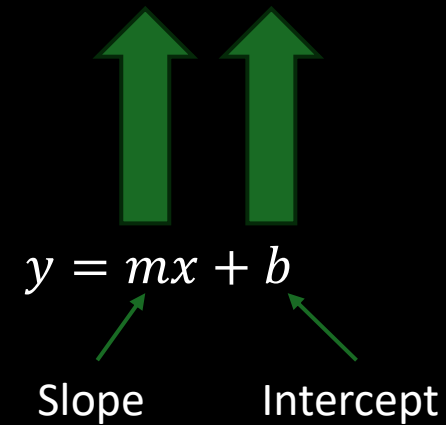
You can guess this trend in your head!



Straight Line With Jitter



```
5 # Straight line with jitter
6 x = jax.random.normal(rng, (100,))
7 jitter = jax.random.normal(new_key, (100,))
8 y = 3*x + 2 + 0.5*jitter
```



How can we determine m and b ?

Straight-line Fit

Analytically...

What are the values of m and b such that we minimize the distance between the line they describe and all the points?

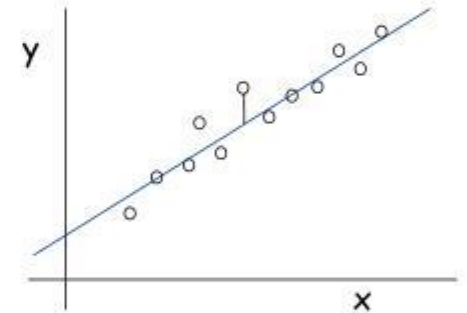
There is an analytical function for this “simple” problem...

Segmented Least Squares

Least squares.

- Foundational problem in statistic and numerical analysis.
- Given n points in the plane: $(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)$.
- Find a line $y = ax + b$ that minimizes the sum of the squared error:

$$SSE = \sum_{i=1}^n (y_i - ax_i - b)^2$$



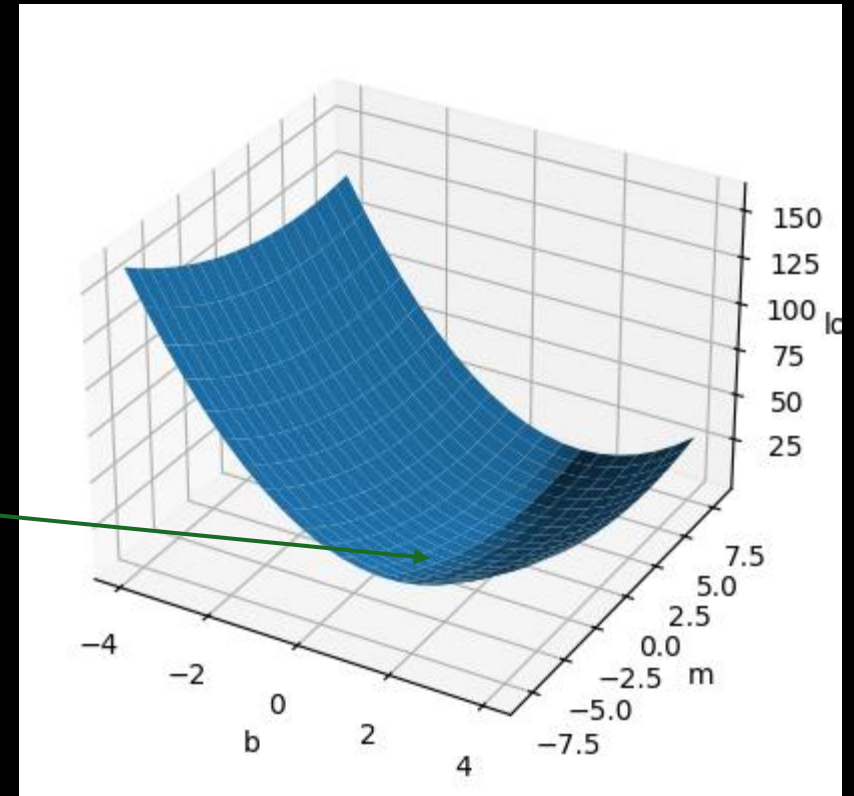
Solution. Calculus \Rightarrow min error is achieved when

$$a = \frac{n \sum_i x_i y_i - (\sum_i x_i) (\sum_i y_i)}{n \sum_i x_i^2 - (\sum_i x_i)^2}, \quad b = \frac{\sum_i y_i - a \sum_i x_i}{n}$$

The Sum Of The Distances...

Plot all the sums of distances for all possible m and b .

We want the minimum value of this plot!

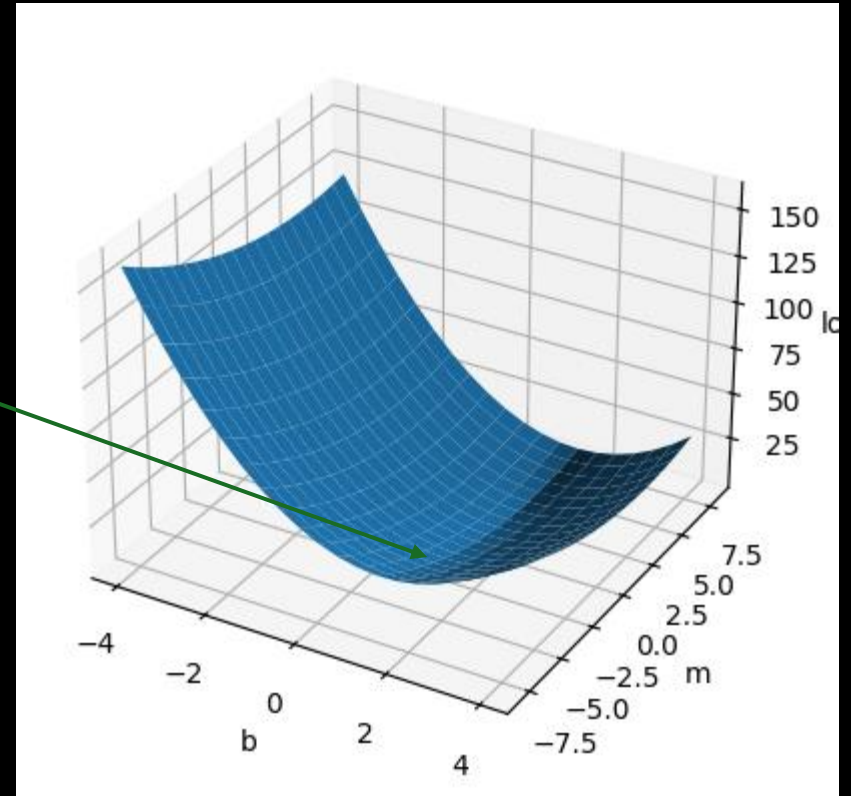


The Sum Of The Distances...

The minimum value

We need a mathematical algorithm
that can “roll down hill”

“gradient decent”

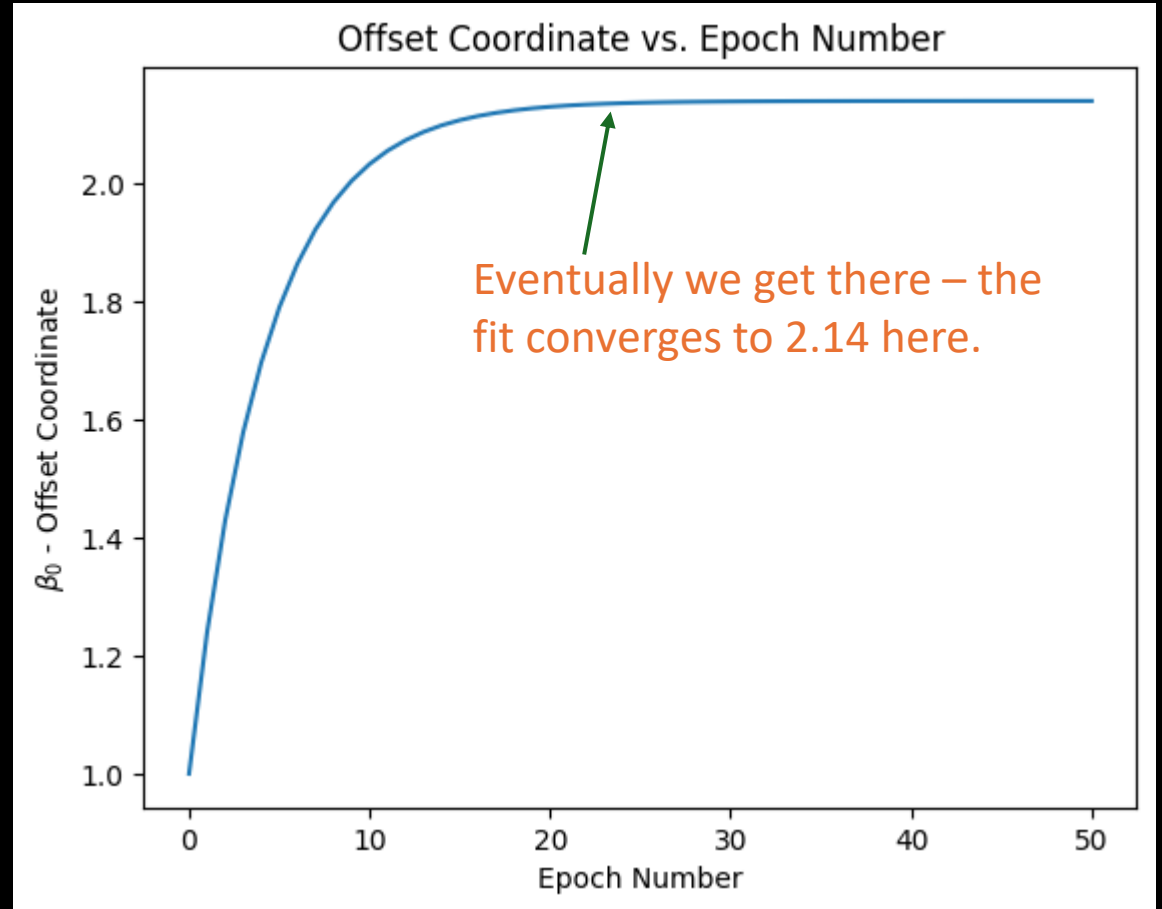


Gradient Decent

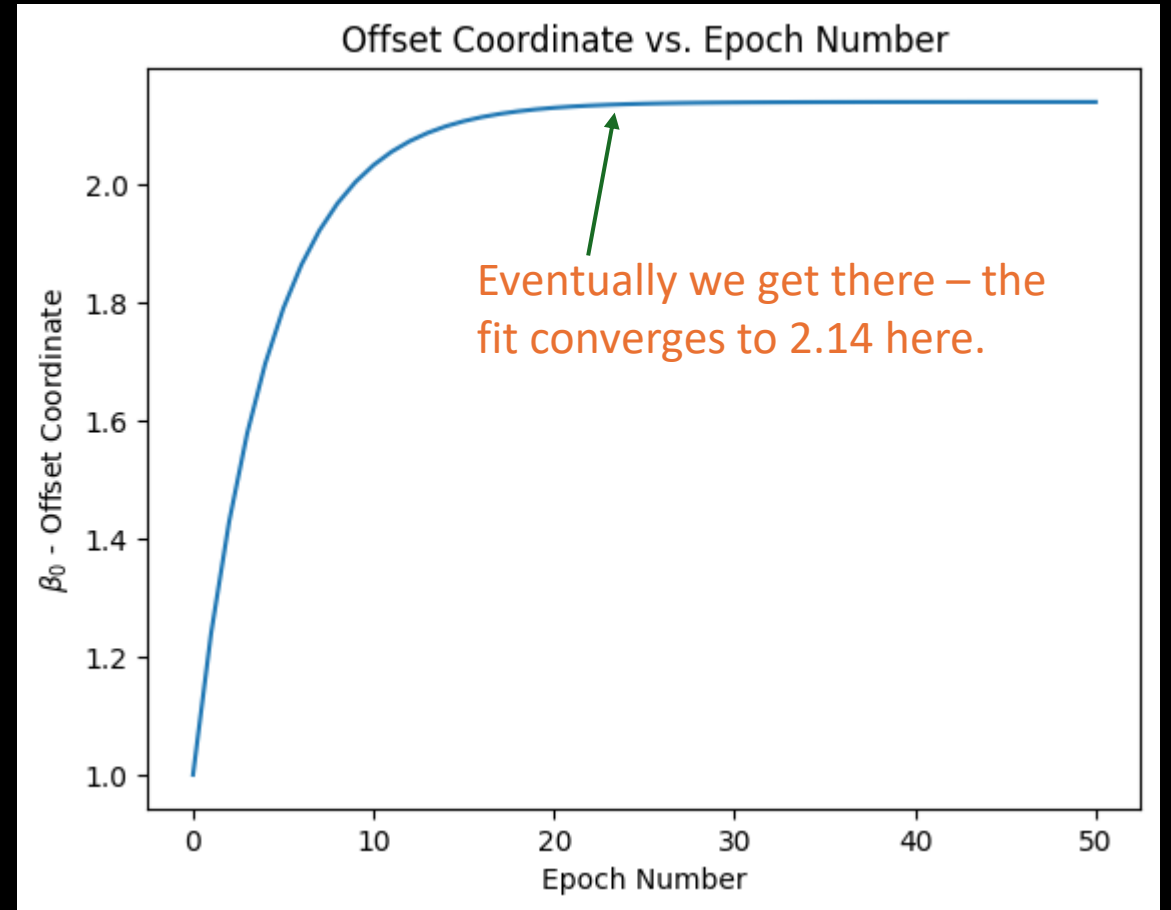
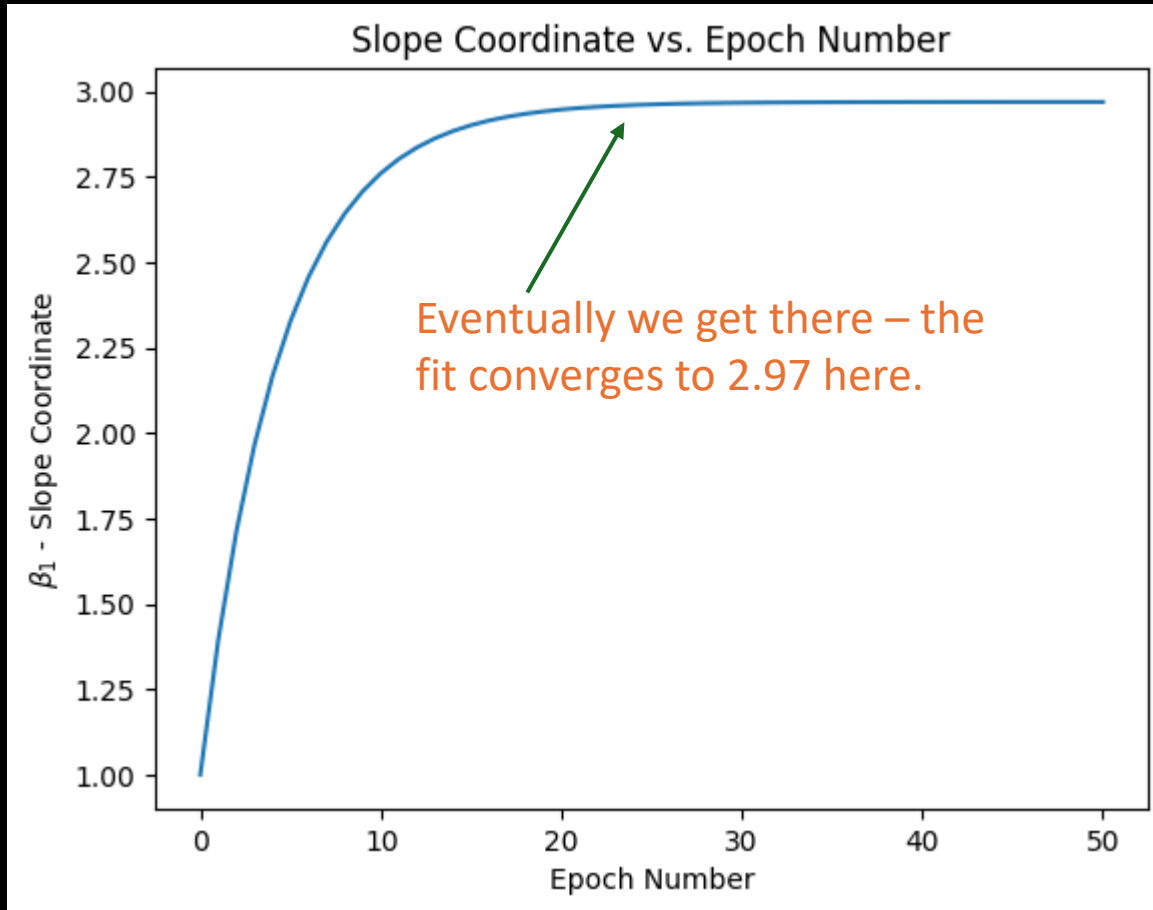
Each *epoch* is a small step down that surface!

But only small steps to as not to “miss” the lowest point.

Recall the target value is 2.0!



Gradient Decent



But that is a straight line...

What is this thing, a Neural Network?

What Is A Neural Network?

A mathematical Function!

Modeled after a biological neuron...

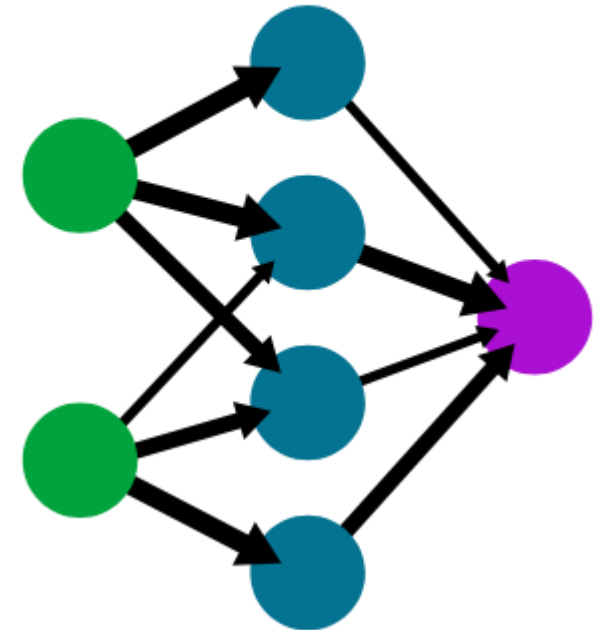
This function has 2 inputs and 1 output...
And has 7 "internal" functions

Numbers flow left-to-right!

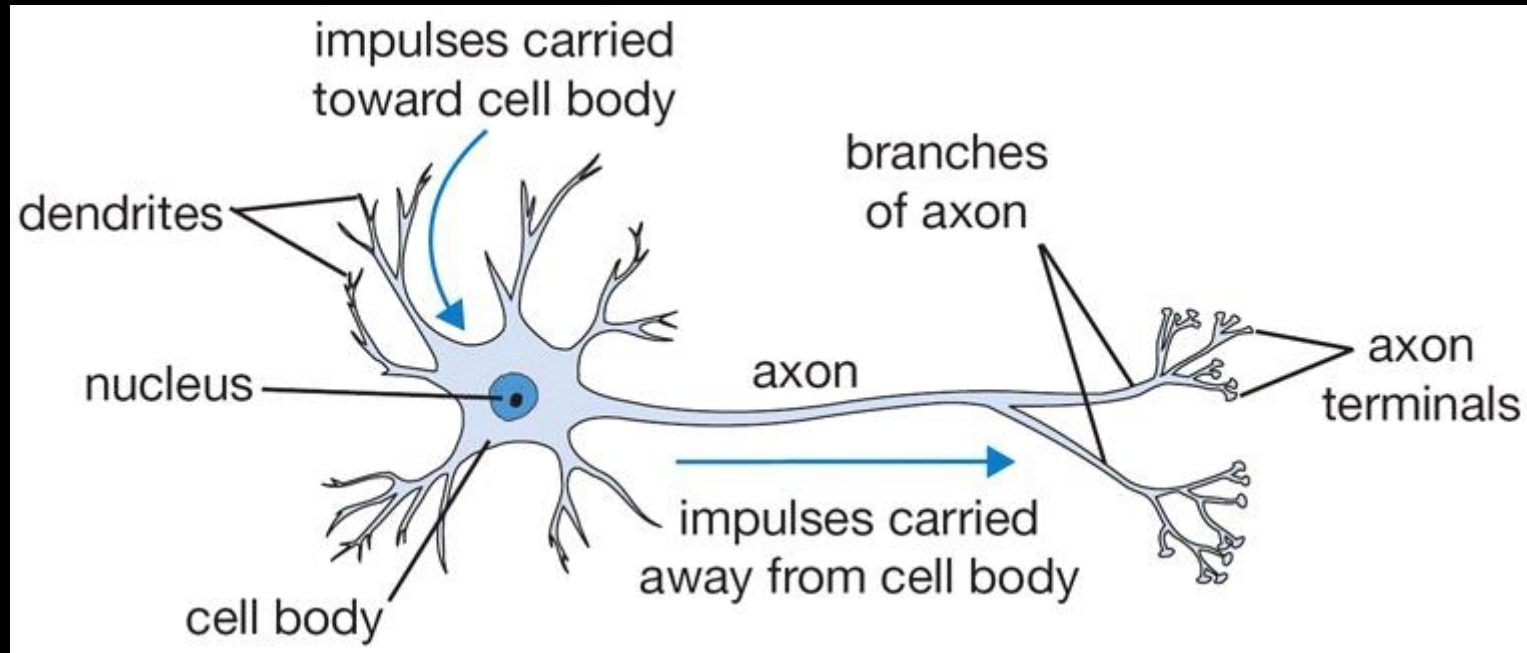


A simple neural network

input layer hidden layer output layer

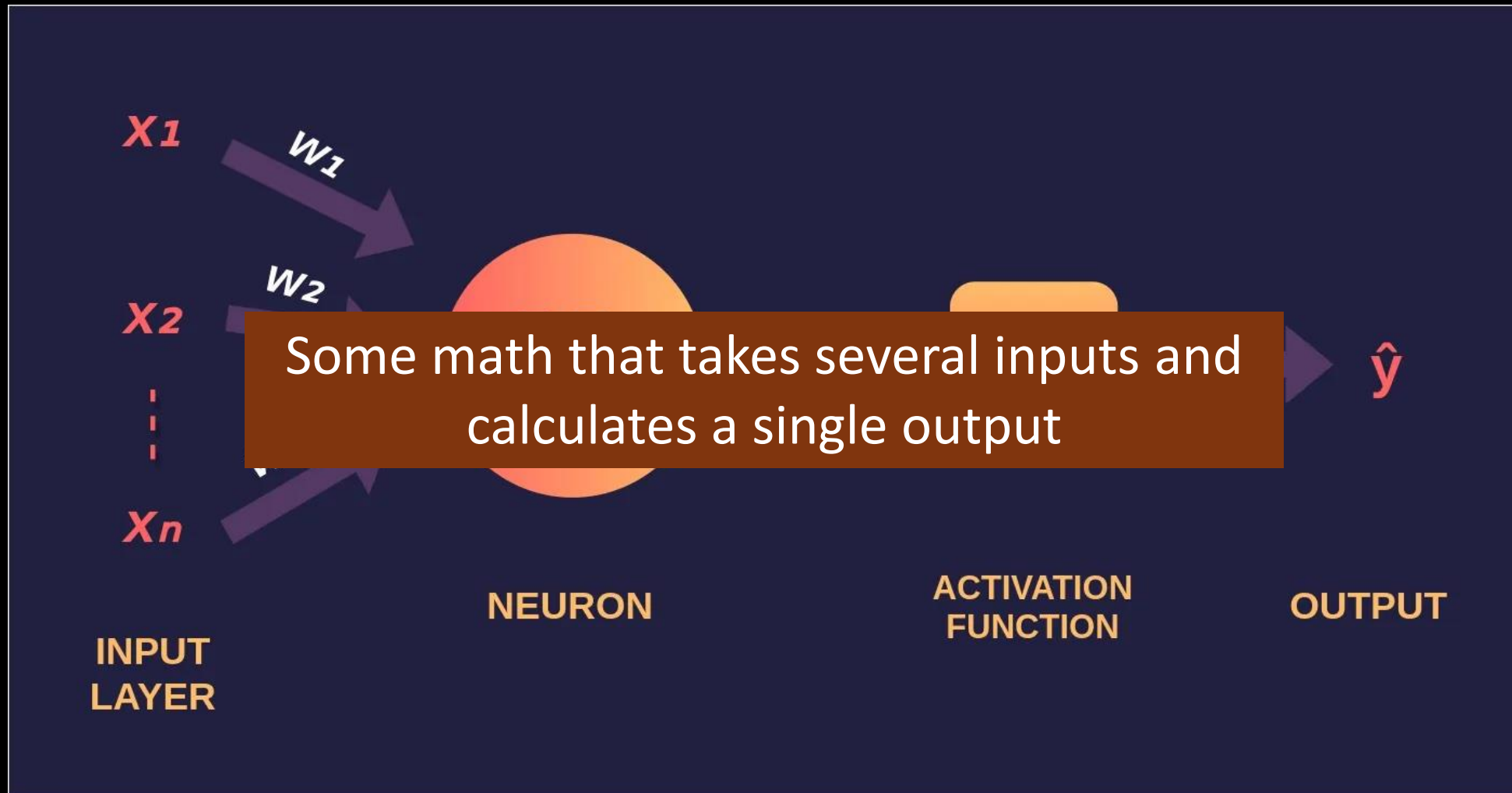


Nuron



(biology)

Nuron?



Ok... skipping many many papers...

1.

If I combine enough of the neurons I can model *any* function I want

2.

I can use my gradient decent from before to tune these functions to model whatever I want

History...

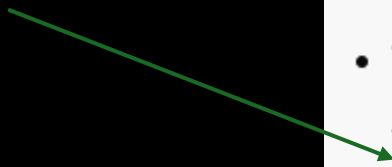
The History of Deep Learning

Below is a timeline of some of the important events in the history of Deep Learning. They are by no means exhaustive, but they are chosen as they help illustrate how Deep Learning arrived to where it is today.


Deep Learning Timeline:


- 1958: Perceptron (Simple version of deep learning) Discovered
- 1979: Convolutional Neural Network Invented (Today dominates image recognition)
- 1982: Recurrent Neural Network Invented (Sequence processing used for NLP)
- ❄️ AI WINTER ❄️
- 2009: ImageNet Introduced

My Dad used simple NN's!



ImageNet (2011)

$f(\text{) = \text{"dog"}$

$f(\text{) = \text{"cat"}$

This is the start of
the Deep Learning
Revolution

Sophistication...

Me: How can I quickly explain deep learning to a lay audience?

Certainly! Let's break down **deep learning** in simple terms for a lay audience:

1. What is Deep Learning?

- Deep learning is a type of **artificial intelligence (AI)** that mimics the way our brains work.
- It involves training **neural networks**, which are like interconnected virtual brain cells, to learn patterns from data.

2. Neural Networks: The Building Blocks

- Imagine a network of interconnected nodes (like neurons in our brain).
- Each node processes information and passes it to others.
- These layers of nodes form a **neural network**.

3. Deep Neural Networks

- “Deep” refers to having many layers in the network.
- Each layer learns progressively complex features.
- The final layer makes predictions (e.g., recognizing cats in images).

4. Training the Network

- We feed the neural network lots of labeled examples (e.g., cat images with labels “cat”).
- It adjusts its internal connections (weights) to get better at making predictions.

5. Applications of Deep Learning

- **Image Recognition:** Identifying objects in photos.
- **Natural Language Processing:** Understanding and generating human language.
- **Recommendation Systems:** Suggesting movies, products, etc.

6. Why “Deep”?

- Deeper networks can learn intricate patterns.
- They're like solving puzzles with more layers.

Remember, deep learning is like teaching a computer to learn from examples, just as we learn from experience! 🤖 🧠

Bing CoPilot
(ChatGPT 4 Turbo)

Deep Learning In Science

Is that Signal?

This was late 1980's

- The tau lepton was of intense interest
 - Still is...
- But we had no large samples of it (then)...

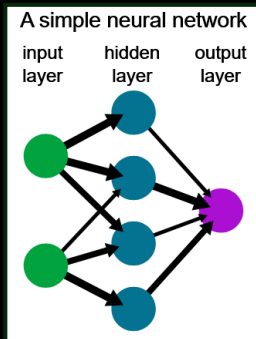
A group, including my Dad, decided to try simple Neural Networks to pick them out of the CDF detector!



Finding Tau's

Built a very simple NN to look for taus.

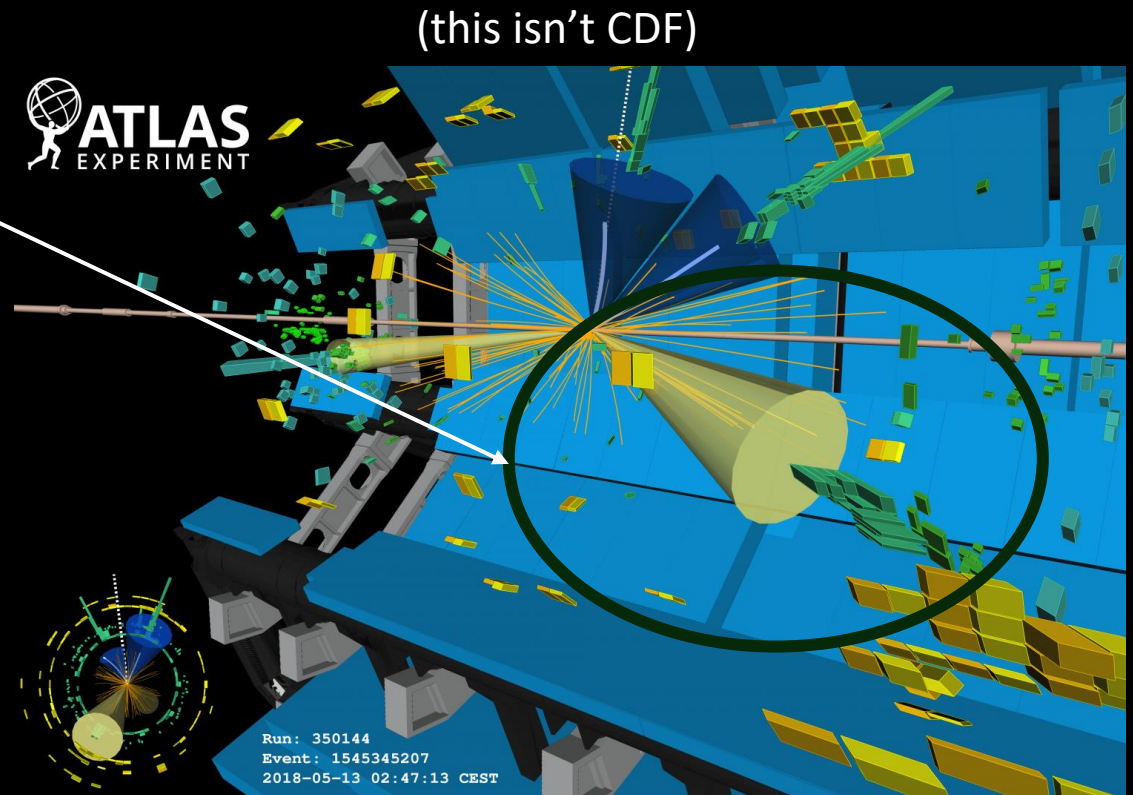
Used a very simple neural network



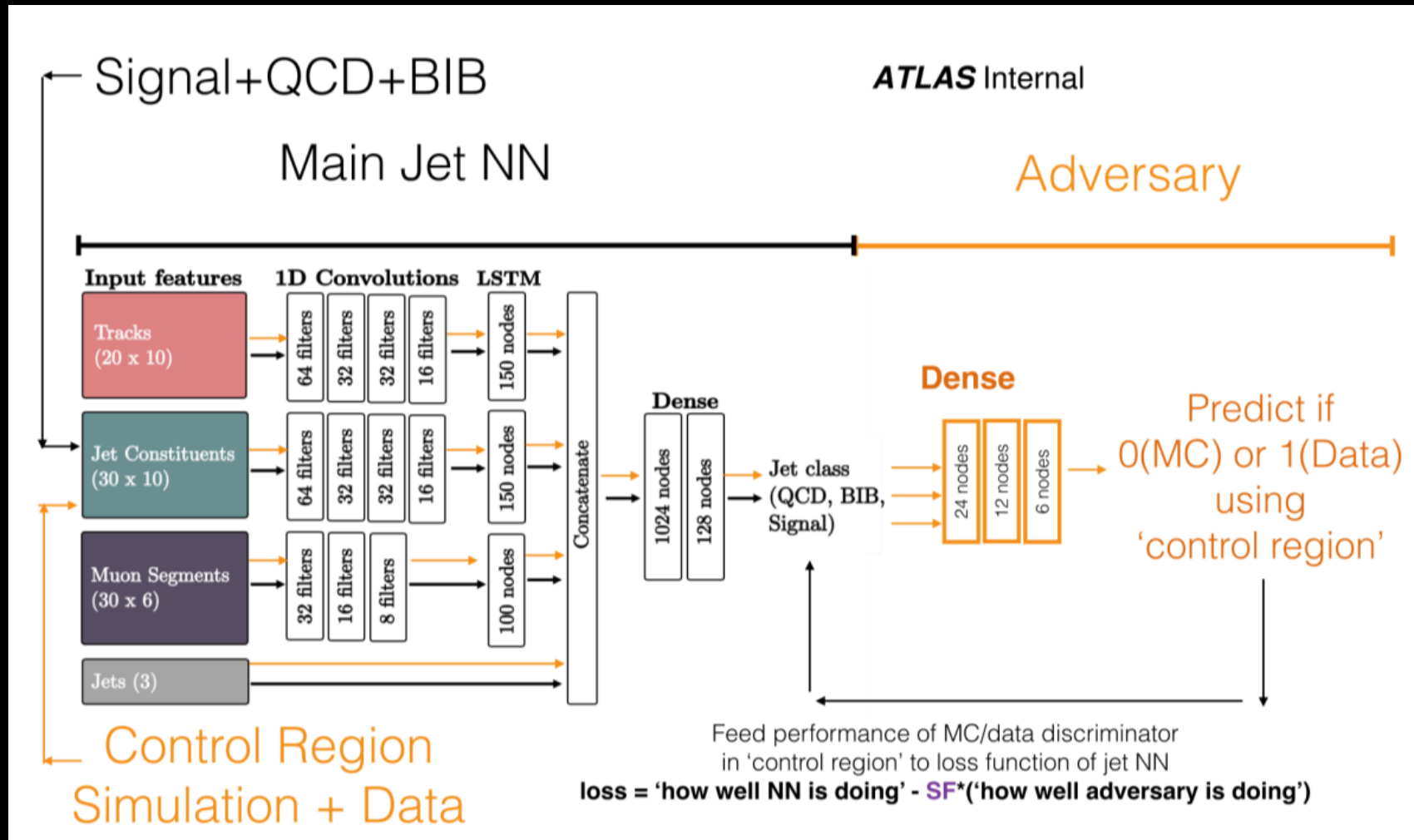
~20 parameters

Meeehhhh... kind-a worked?

Simpler techniques worked better...

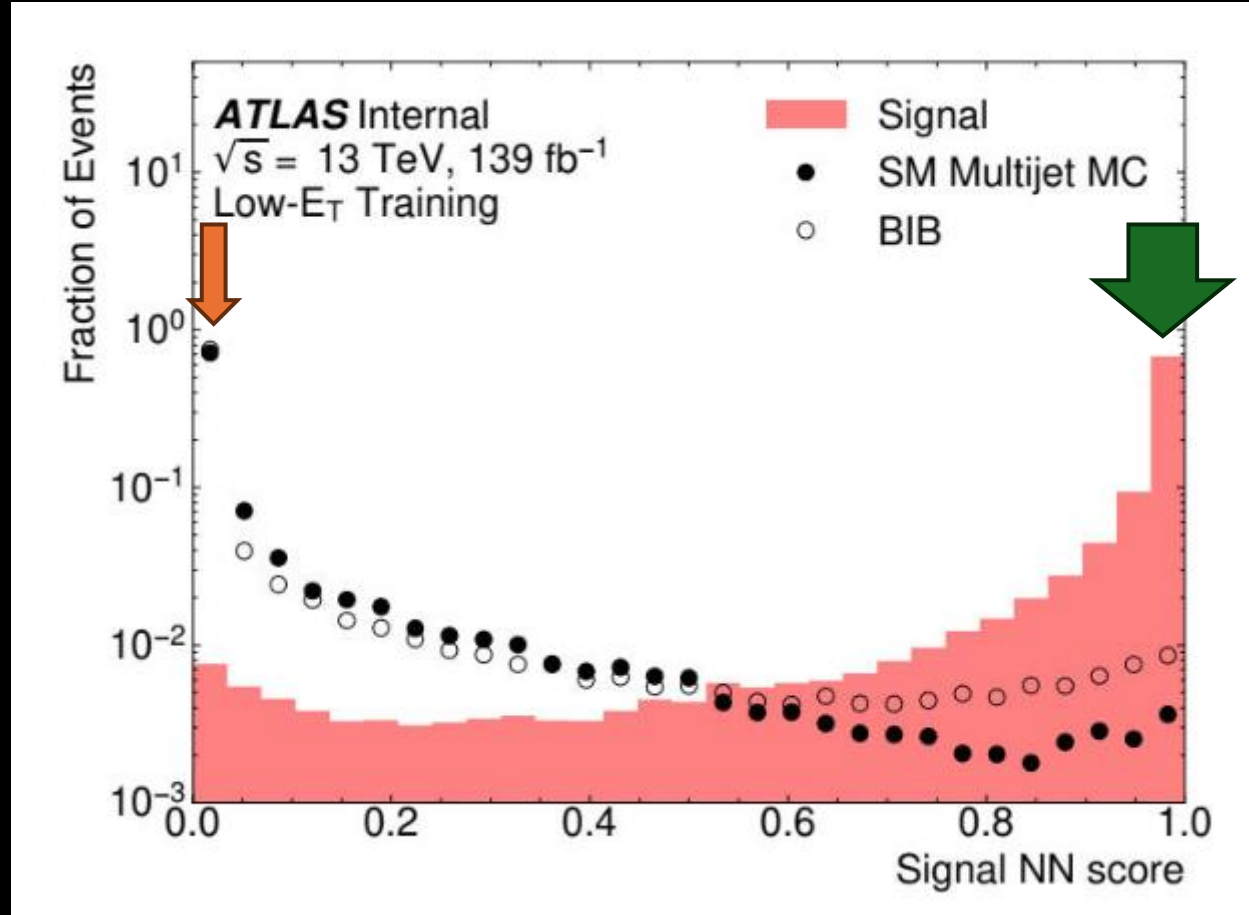


My last published paper...



It works well!

Most of my background is here!

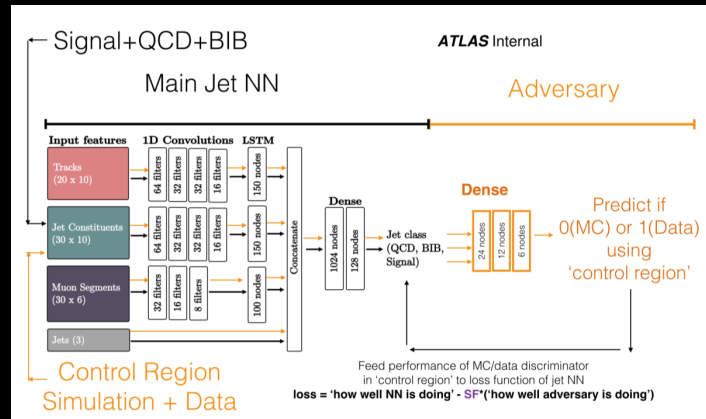


Most of my signal is here!

What is deep learning/Neural Networks/ML?

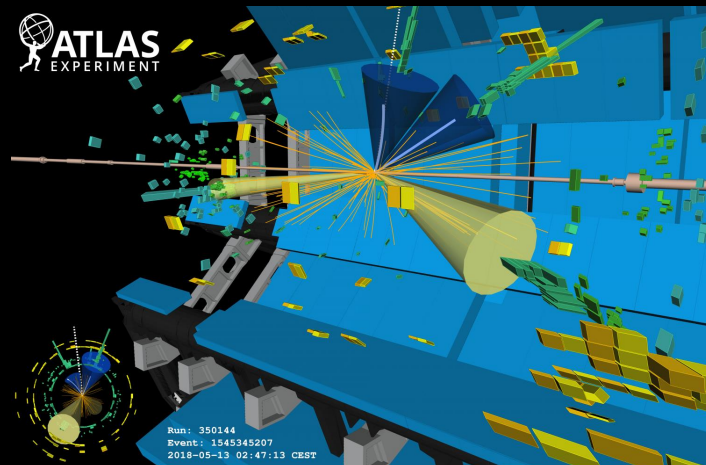
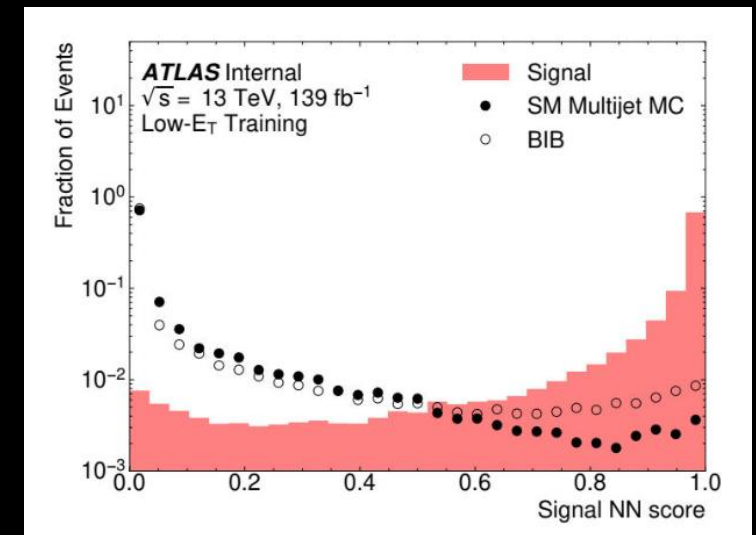
It is function fitting!

Function Fitting



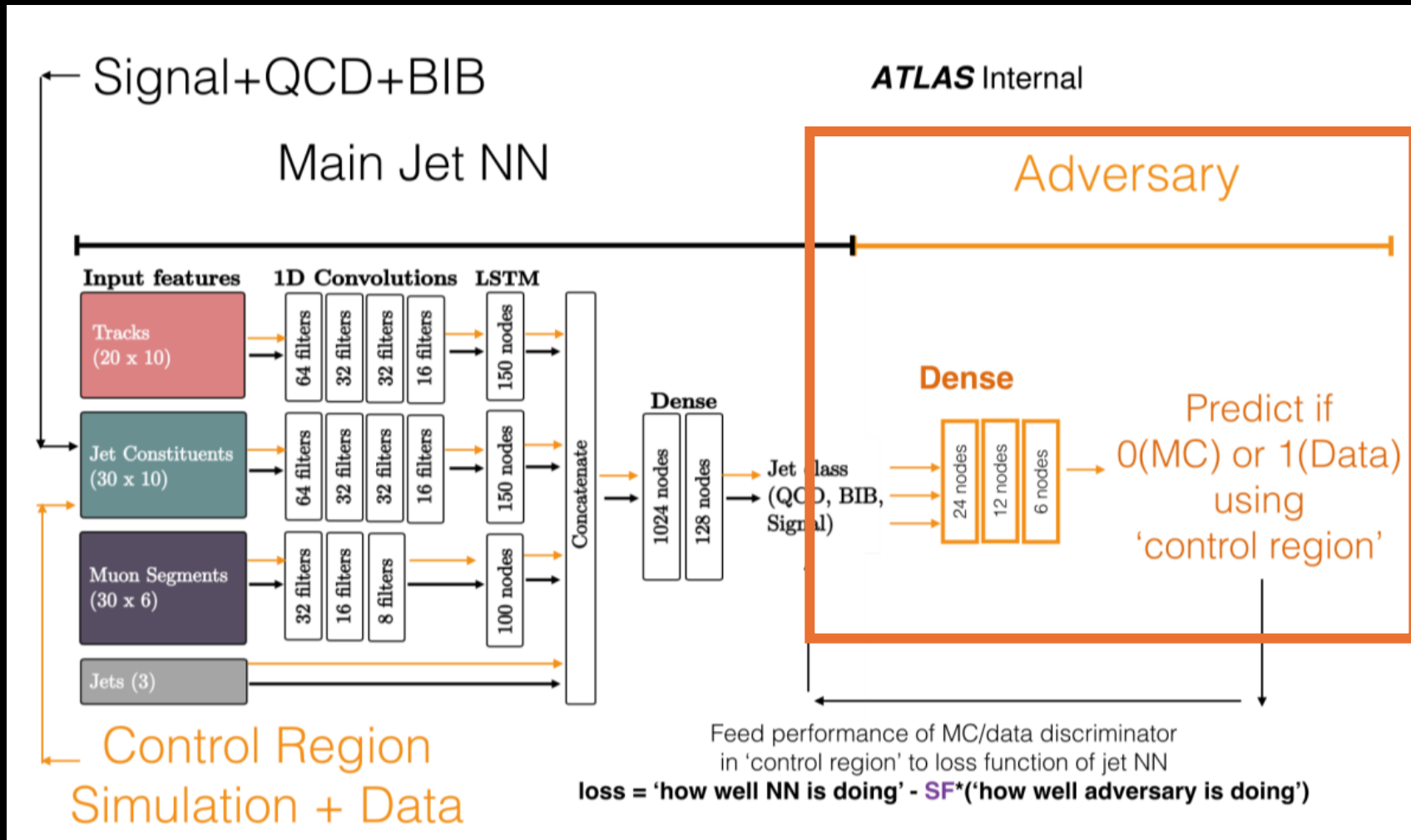
This is my (crazy) function

This is the result of the function
(on many pieces of data)

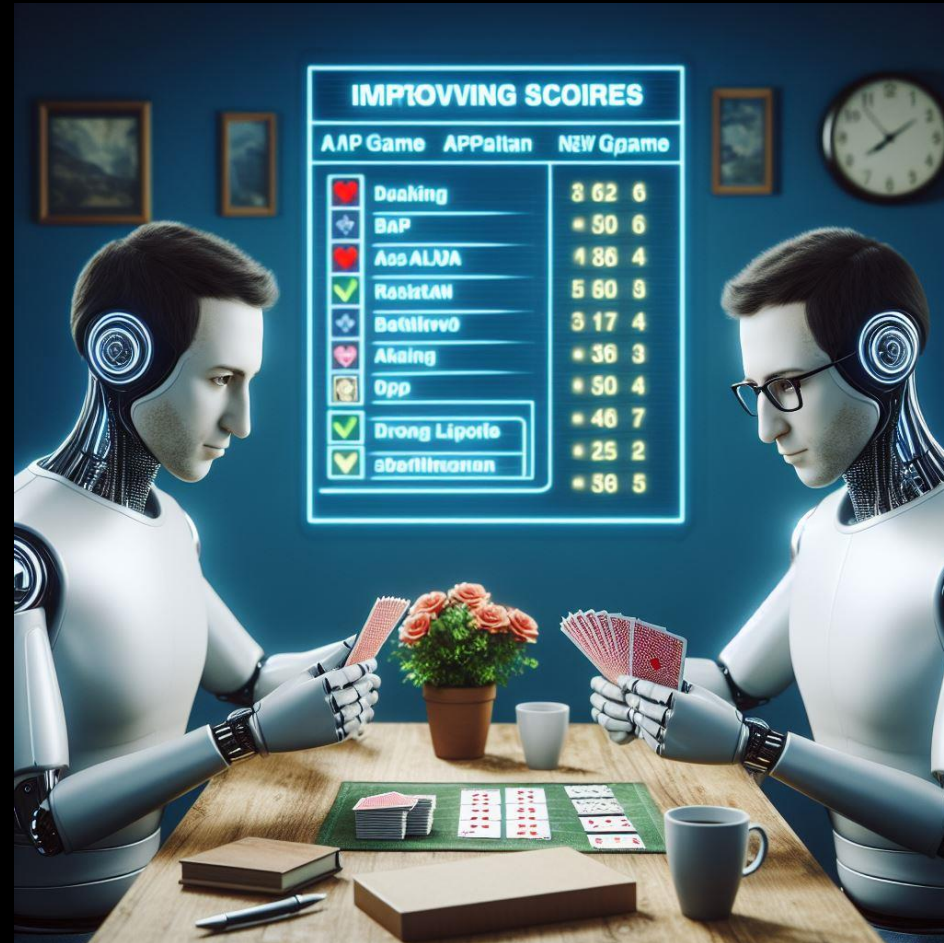


This is one event of my data
(Argument to the function)

How Interesting Are Those Function?



How Interesting Are Those Functions?



This is where Modern Research Is

Training Is Expensive
Data is limited

Can you purpose design a function to accomplish
the task at hand?

This would short-cut your training!

YES

This is the future of LLM's...

arXiv:1706.03762v7 [cs.CL] 2 Aug 2023

Provided proper attribution is provided, Google hereby grants permission to reproduce the tables and figures in this paper solely for use in journalistic or scholarly works.

Attention Is All You Need

Ashish Vaswani* Google Brain avaswani@google.com	Noam Shazeer* Google Brain noam@google.com	Niki Parmar* Google Research nikip@google.com	Jakob Uszkoreit* Google Research usz@google.com
Llion Jones* Google Research llion@google.com	Aidan N. Gomez*[†] University of Toronto aidan@cs.toronto.edu	Lukasz Kaiser* Google Brain lukaszkaizer@google.com	
Illia Polosukhin*[‡] illia.polosukhin@gmail.com			

Abstract

The dominant sequence transduction models are based on complex recurrent or convolutional neural networks that include an encoder and a decoder. The best performing models also connect the encoder and decoder through an attention mechanism. We propose a new simple network architecture, the Transformer, based solely on attention mechanisms, dispensing with recurrence and convolutions entirely. Experiments on two machine translation tasks show these models to be superior in quality while being more parallelizable and requiring significantly less time to train. Our model achieves 28.4 BLEU on the WMT 2014 English-to-German translation task, improving over the existing best results, including ensembles, by over 2 BLEU. On the WMT 2014 English-to-French translation task, our model establishes a new single-model state-of-the-art BLEU score of 41.8 after training for 3.5 days on eight GPUs, a small fraction of the training costs of the best models from the literature. We show that the Transformer generalizes well to other tasks by applying it successfully to English constituency parsing both with large and limited training data.

^{*}Equal contribution. Listing order is random. Jakob proposed replacing RNNs with self-attention and started the effort to evaluate this idea. Ashish, with Illia, designed and implemented the first Transformer models and has been crucially involved in every aspect of this work. Noam proposed scaled dot-product attention, multi-head attention and the parameter-free position representation and became the other person involved in nearly every detail. Niki designed, implemented, tuned and evaluated countless model variants in our original codebase and tensor2tensor. Llion also experimented with novel model variants, was responsible for our initial codebase, and efficient inference and visualizations. Lukasz and Aidan spent countless long days designing various parts of and implementing tensor2tensor, replacing our earlier codebase, greatly improving results and massively accelerating our research.

[†]Work performed while at Google Brain.
[‡]Work performed while at Google Research.

31st Conference on Neural Information Processing Systems (NIPS 2017), Long Beach, CA, USA.

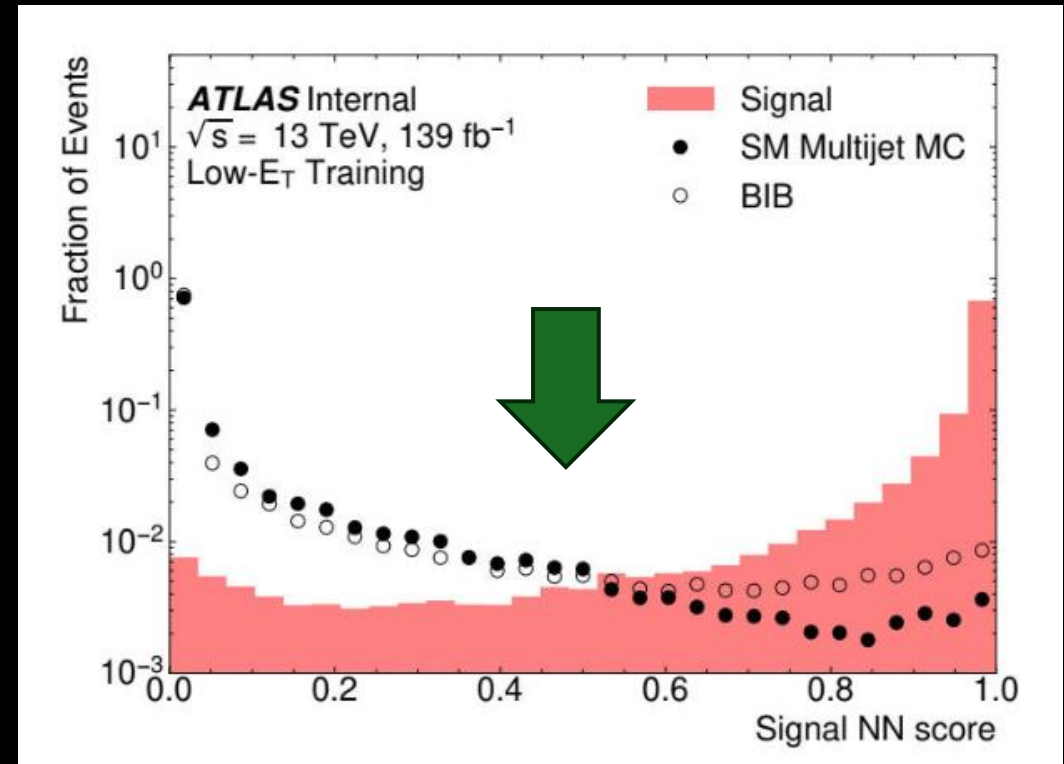
Large Language Models

MOAR DATA!!!

For this analysis producing data was expensive

- Time
- Compute time

More data would likely have meant better separation!



Lawsuits!

A friend of mine who works on large language models at Meta:

“No LLM can be trained without using Copyrighted data. There just isn’t enough high quality data to train on.”

Right now: No one is making money on LLM’s/AI

The Times Sues OpenAI and Microsoft Over A.I. Use of Copyrighted Work

Millions of articles from The New York Times were used to train chatbots that now compete with it, the lawsuit said.

[Share full article](#) [Share](#) [Bookmark](#) [1.3K](#)



A lawsuit by The New York Times could test the emerging legal contours of generative A.I. technologies. Sasha Maslov for The New York Times

  **By Michael M. Grynbaum and Ryan Mac**

Dec. 27, 2023

[Leer en español](#)

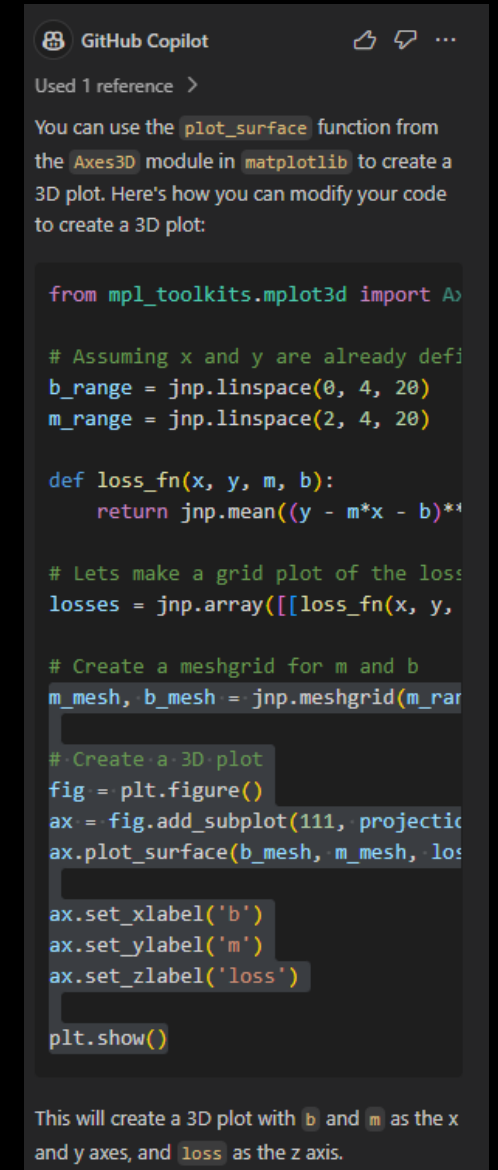
The New York Times sued OpenAI and Microsoft for copyright infringement on Wednesday, opening a new front in the increasingly intense legal battle over the unauthorized use of published work to train artificial intelligence technologies.

Can LLM's Help In Science?

They already are:

- Writing first draft of code to process our data
- General lookup questions
 - But you sort-of have to know the answers
- Helping to smooth our poor English...

...as assistants



The screenshot shows the GitHub Copilot interface. At the top, it says "GitHub Copilot" with a logo and some icons. Below that, it says "Used 1 reference >". The main text says: "You can use the `plot_surface` function from the `Axes3D` module in `matplotlib` to create a 3D plot. Here's how you can modify your code to create a 3D plot:"

```
from mpl_toolkits.mplot3d import Axes3D

# Assuming x and y are already defined
b_range = jnp.linspace(0, 4, 20)
m_range = jnp.linspace(2, 4, 20)

def loss_fn(x, y, m, b):
    return jnp.mean((y - m*x - b)**2)

# Lets make a grid plot of the losses
losses = jnp.array([[loss_fn(x, y, m, b) for m in m_range] for y in b_range])

# Create a meshgrid for m and b
m_mesh, b_mesh = jnp.meshgrid(m_range, b_range)

# Create a 3D plot
fig = plt.figure()
ax = fig.add_subplot(111, projection='3d')
ax.plot_surface(b_mesh, m_mesh, losses)

ax.set_xlabel('b')
ax.set_ylabel('m')
ax.set_zlabel('loss')

plt.show()
```

At the bottom, it says: "This will create a 3D plot with `b` and `m` as the x and y axes, and `loss` as the z axis."

But much more imaginative uses are coming...

1. Physicist must write talk night before conference presentation.
2. Physicist is bored...
3. Writes a prompt and feeds it and a conference abstract to a LLM
4. Discovers it can tell them if they are interested in the conference talk!
5. Writes a python script to feed all the conference talks to the LLM and rank them for interest!

This was 3 hours of coding, starting from not really knowing how to do this...

[From github...](#)

```
data = load_indico_json(event_url)
prompt = """I am an expert in experimental particle physics as well as computing for
particle physics. You are my expert AI assistant who is well versed in particle physics
and particle physics computing. My interests are in the following areas:
1. Hidden Sector Physics
2. Long Lived Particles (Exotics or RPV SUSY)
3. Analysis techniques and methods and frameworks, particularly those based around python or ROOT's DataFrame (RDF)
4. Machine Learning and AI for particle physics
5. Distributed computing for analysis (e.g. Dask, Spark, etc)
6. Data Preservation and FAIR principles
7. Differentiable Programming

I'm not very interested in:
1. Quantum Computing
2. Lattice QCD
3. Neutrino Physics

Please summarize this conference abstract so I can quickly judge the abstract and if I want to
see the talk it represents.

Your reply should have the following format:

summary: <One line, terse, summary of the abstract that does not repeat the title. It should add extra information beyond
experiment: <If you can guess the experiment this abstract is associated with (e.g. ATLAS, CMS, LHCb, etc), place it here
keywords: <comma separated list of keywords that match my interest list above. If you can't find any, leave blank.>
interest: <If you can guess how interested I am from above, put "low", "medium", or "high" here. Otherwise blank.>

Here is the talk title and Abstract:"""
```

LLM for Particle Physics?

Question: Do [large language] models actually understand what they are saying?

“Clearly, some people believe they do, and some people believe they are just stochastic parrots.”

Geoff Hinton

Wait... what is a LLM? What is it actually trained to do!???

Training Is... Simple...

Artificial intelligence seems more powerful than ever, with chatbots like Bard and ChatGPT capable of producing uncannily humanlike text.

Source Text (copyrighted?)

Artificial intelligence seems more powerful than ever, with chatbots like Bard and ChatGPT capable of producing uncannily humanlike

Training Input


text

Expected Output

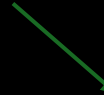
They are trained to predict the next word....

That is all

Stochastic Parrot

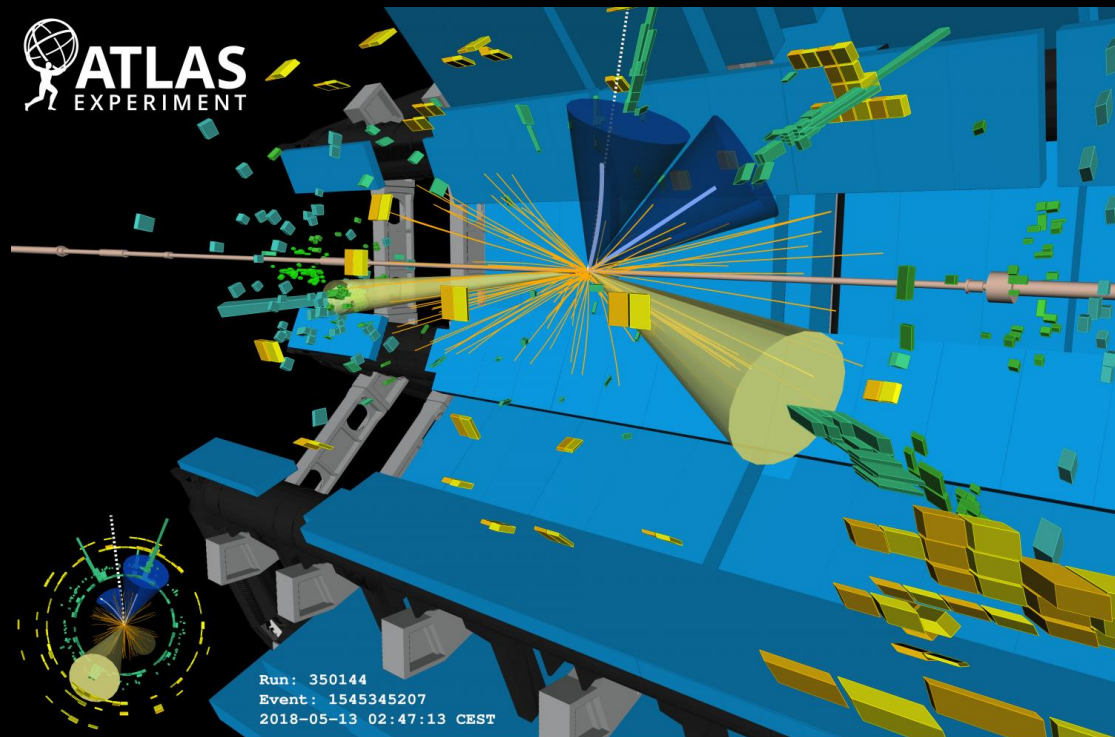


There is some randomness injected
into the process



It was trained to complete a
sentence! Sheesh!

For Science!



- We could train a NN to know how a detector works
- Train on all our Simulated Data or even Real Data
 - It would learn about objects
 - How objects occur together

Then we could tune it to answer specific questions...

This effort is just starting (funding!!!)

Will It Work?

I can't wait to find out!

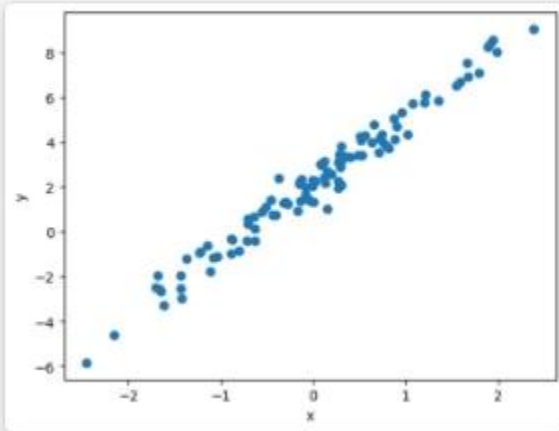


But... we have a way to go...

Ask Copilot (ChatGPT 4 turbo?)...

I had to convince it to try – as if it knew it couldn't do a good job...

Could you tell me if this looks like it is a straight line, and then estimate the slope and intercept?



Certainly! The scatter plot in the image appears to form a **straight ascending line**. Let's estimate the slope and intercept:

- **Slope (m):** The rise from left to right suggests a positive slope.
- **Intercept (b):** The line intersects the y-axis slightly above 0.

Given this, we can approximate the equation of the line as:

$$y = mx + b$$

However, without specific data points or coordinates, I cannot provide precise values for the slope and intercept. If you have the data, I'd be happy to calculate them for you!

Copilot

Certainly! Let's analyze the scatter plot to estimate the slope and intercept.

1. **Slope (m):** The line rises from left to right, indicating a positive slope. To estimate the slope, we can visually approximate the change in y-values divided by the change in x-values. Let's select two points on the line:

- Point A: Approximately (-2, -6)
- Point B: Approximately (2, 8)

The change in y-values is $8 - (-6) = 14$, and the change in x-values is $2 - (-2) = 4$. Therefore, the slope (m) can be estimated as: $[m \approx \frac{\{\Delta y\}}{\{\Delta x\}} = \frac{\{14\}}{\{4\}} = 3.5]$

2. **Intercept (b):** The line intersects the y-axis slightly above 0. Let's estimate the y-intercept visually. It appears to be around $(y = 4)$.

Therefore, the equation of the line can be approximated as: $[y = 3.5x + 4]$

Keep in mind that the inspection. If you have exact slope and intercept. 😊

Close, but a ways to go...

$$y = 3x + 2$$

Machine Learning sucks! (compared to humans and animals)

- ▶ Supervised learning (SL) requires large numbers of labeled samples.
- ▶ Reinforcement learning (RL) requires insane amounts of trials.
- ▶ Self-Supervised Learning (SSL) works great but...
 - ▶ Generative prediction only works for text and other discrete modalities

- ▶ **Animals and humans:**
 - ▶ Can learn new tasks **very** quickly.
 - ▶ Understand how the world works
 - ▶ Can reason an plan
- ▶ **Humans and animals have common sense**
- ▶ **There behavior is driven by objectives (drives)**

Future AI Assistants need Human-Level AI

- ▶ **AI assistants will require (super-)human-level intelligence**
 - ▶ Like having a staff of smart “people” working for us
- ▶ **But, we are nowhere near human-level AI today**
 - ▶ Any 17 year-old can learn to drive in 20 hours of training
 - ▶ Any 10 year-old can learn to clear the dinner table in one shot
 - ▶ Any house cat can plan complex actions
- ▶ **What are we missing?**
 - ▶ Learning how to world works (not just from text)
 - ▶ World models. Common sense
 - ▶ Memory, Reasoning, Hierarchical Planning

Fin!

- The only AI game in town is Machine Learning
- Machine Learning is just a function fit
- It has had a huge impact on Science
 - Which started about a decade ago
- Is now having a huge impact on Society for the same reasons

