

# Intro to Neural Nets

Course Logistics and Introduction

# Today's Agenda

## 1. COURSE LOGISTICS

- Website, schedule, grading and evaluation criteria.
- Course textbook, lecture format, etc.

## 2. INTERESTING USE CASES

- Frivolous, academic, and practically useful.
- A recent failure, and societal concerns.

## 3. QUICK INTRODUCTION

- What is a neural network?
- How does it work?





accenture



# About Me



**MSI** at the  
**ARF**

 **FFG**  
Forschung wirkt.

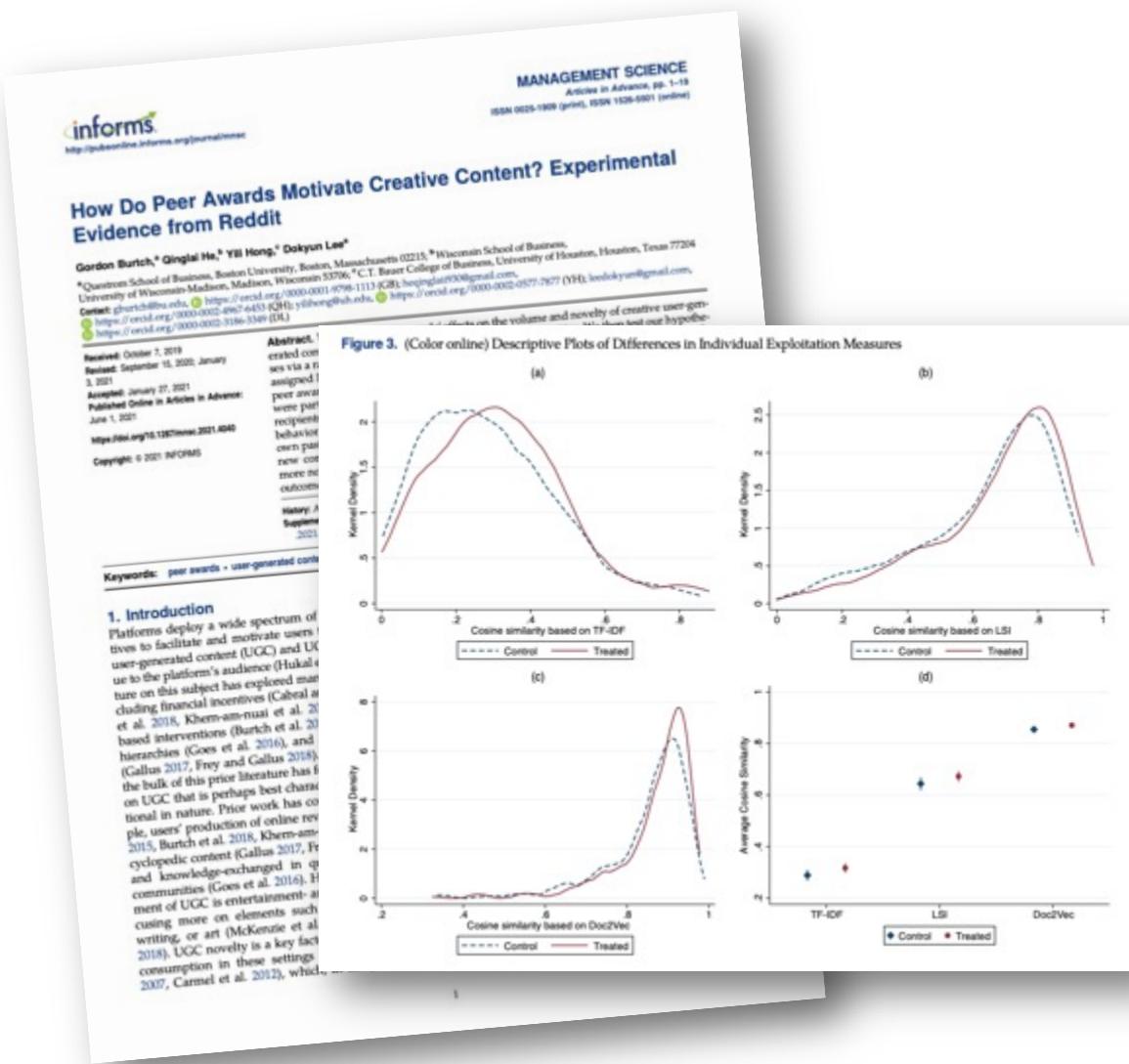


 **Adobe**



 **Microsoft**

# My Research



## Achieving Reliable Causal Inference with Data-Mined Variables: A Random Forest Approach to the Measurement Error Problem

Mochen Yang<sup>1</sup>, Edward McFowland III<sup>1</sup>, Gordon Burtch<sup>1</sup>, Gediminas Adomavicius<sup>1</sup>

<sup>1</sup>University of Minnesota, Carlson School of Management

March 13, 2020

### Abstract

Combining machine learning with econometric analysis is becoming increasingly prevalent in both research and practice. A common empirical strategy involves the application of predictive modeling techniques to "mine" variables of interest from available data, followed by the inclusion of those variables into an econometric framework, with the objective of estimating causal effects. Recent work highlights that, because the predictions from machine learning models are inevitably imperfect, econometric analyses based on the predicted variables are likely to suffer from bias due to measurement error. We propose a novel approach to mitigate these biases, leveraging the ensemble learning technique known as the random forest. We propose employing random forest not just for prediction, but also for generating instrumental variables to address the measurement error embedded in the prediction. The random forest algorithm performs best when comprised of a set of trees that are individually accurate in their predictions, yet which also make "different" mistakes, i.e., have weakly correlated prediction errors. A key observation is that these properties are closely related to the relevance and exclusion requirements of valid instrumental variables. We design a data-driven procedure to select tuples of individual trees from a random forest, in which one tree serves as the endogenous covariate and the other trees serve as its instruments. Simulation experiments demonstrate the efficacy of the proposed approach in mitigating estimation biases, and its superior performance over three alternative methods for bias correction.

**Keywords:** machine learning, econometric analysis, instrumental variable, random forest, causal inference

### 1 Introduction

Advances in predictive machine learning have enabled researchers to extract useful information from various types of data, such as text and images, which would otherwise be difficult or costly to codify at scale. For example, recent academic work has highlighted that cutting-edge prediction techniques are now capable of inferring the socioeconomic properties of a localized population (e.g., income/social distribution) from the models and masks of cars appearing in Google Street View images (Gohru et al., 2017) and detecting adverse drug events based on drug attributes (Ryu et al., 2018). These measurements, now available at scale and with little cost, can enable empirical investigations of important questions in economics, health care, and many other domains.

Indeed, many researchers have begun doing exactly that, first using predictive machine learning to construct or populate a variable of interest, e.g., using text mining tools to predict text sentiment, and then including that variable into econometric models as an independent covariate. This practice has become prevalent in multiple social science domains, including economics (Jelveh et al., 2015), political science (Fog and Tyler, 2017), and management (Yang et al., 2018).

# Course Materials

## COURSE WEBSITE

- The course website is on Blackboard – I assume you have looked at it by this point (if you aren't enrolled in the site, let me know)
- You will submit all assignments and receive relevant course announcements via this site.
- I will post lecture materials and in-class exercises / examples in the GitHub Repository that is linked on Blackboard.

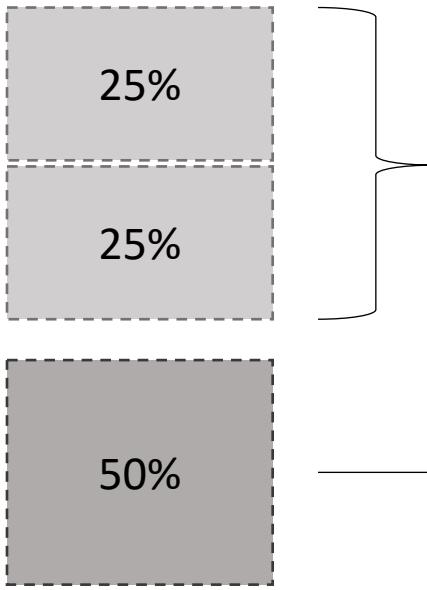
The screenshot shows the Blackboard course homepage for BA865 Advanced Analytics - Intro to Neural Nets (Spring 2022). The left sidebar includes links for Roster, Course Description, Progress Tracking, Attendance, Groups, Announcements, and Books & Tools. The main content area displays the 'Course Content' section, which features a link to the 'Course Materials (GitHub Repo)'. A note states: 'All course materials will be distributed via the below GitHub repository. Check here for latest version of the syllabus, slides, code examples, etc.' Below this is a section for the 'BA865 GitHub Repo', with a note: 'This GitHub repository will remain up-to-date with all relevant slides, examples, and datasets.' There are also sections for 'Assignments' and 'Quizzes / Exams'.



## GOOGLE COLAB

- All homework and exercises in this course will be implemented in Python. You are welcome to employ a local instance of Python, but for simplicity's sake I would recommend you work in Google Colab, because we (the TA and I) will not be able to provide technical support. We will rely heavily on Jupyter Notebooks.

# Grading and Evaluation



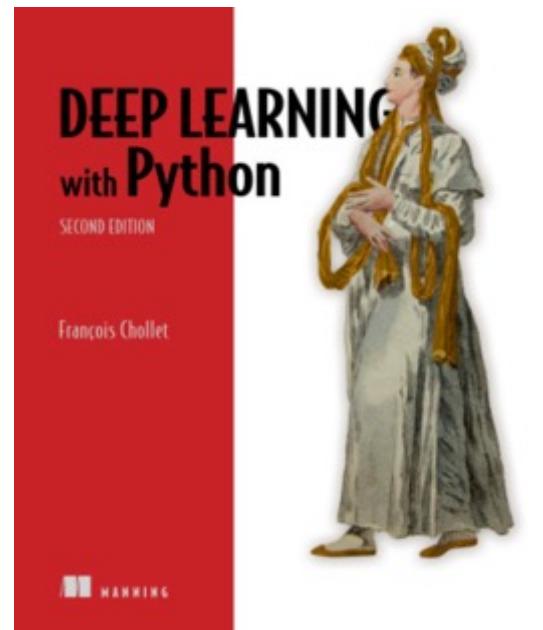
## HOMEWORK

- Two (2) assignments worth 25% of your final grade, each, to be completed in groups of no more than 2 students (indicate group members as part of your submission); you can also complete these assignments by yourself, if you prefer.
- Each homework problem set is based on preceding lab / course material. Due before the start of class on the indicated submission date – **submit via Blackboard only!**
- Per the syllabus, late submissions will not be accepted.

## FINAL PROJECT

- The final project will be presented during the last regularly scheduled classroom session.
- You will work in groups of 2-3 (your choice) to implement a neural network-based predictive model that addresses a practical problem of interest to you! You will source your own dataset, motivate the prediction problem (explain why it's practically relevant), and then implement the predictive model. Your goal should be to make this project useful for you!
- The deliverables will include a project proposal due in Week 2 (5%), a mid-point check-in meeting with me to ensure you are on track (5%)

# Course Textbook



Chollet, François. (2021). *Deep Learning with Python (2<sup>nd</sup> Edition)*.  
Manning Publications Co. **ISBN-13: 978-1617296864**.  
<https://www.manning.com/books/deep-learning-with-python-second-edition>

# Required Software

## SOFTWARE CONFIGURATION

- You can access Google Colab at <https://colab.research.google.com>. You probably want to use your BU Google account credentials.
- If you wish, you may run Python / Jupyter notebooks locally, installing the relevant packages, but please note that we (the TA and I) cannot provide any technical support.



# Why Keras?

Figure 1.12 Machine learning tools used by top teams on Kaggle

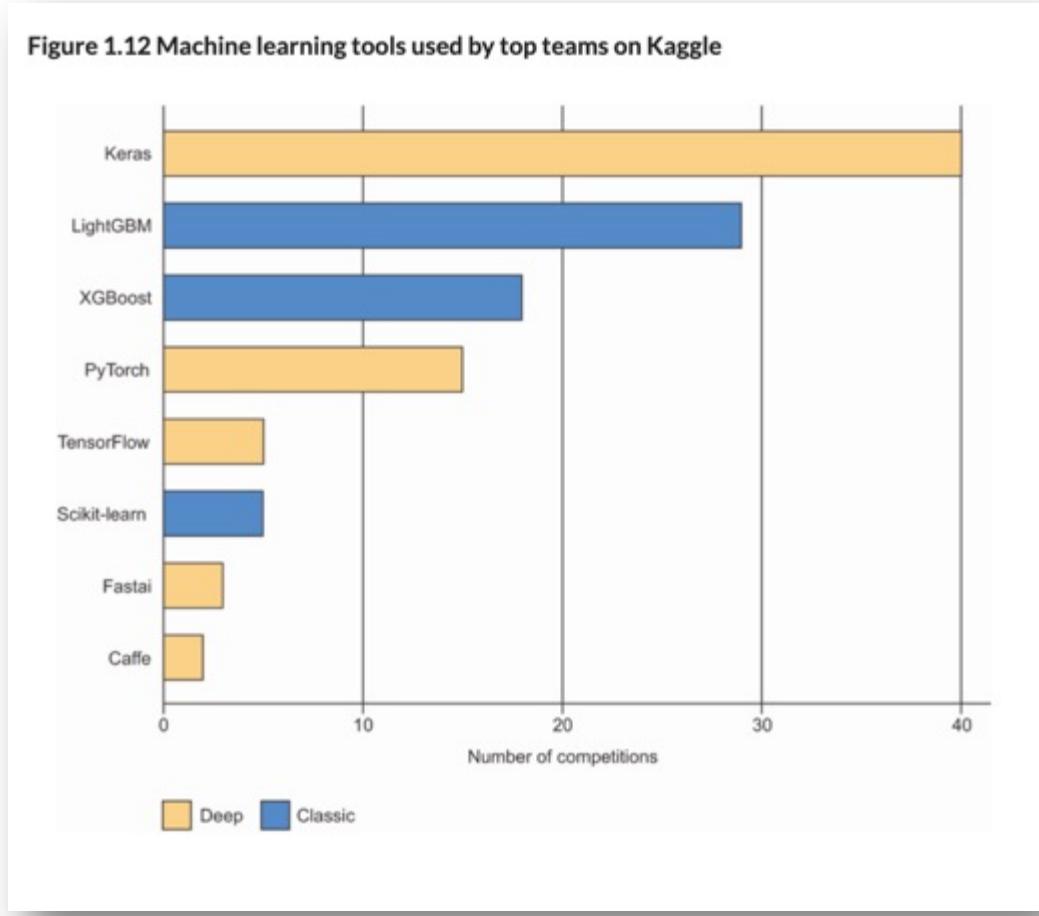
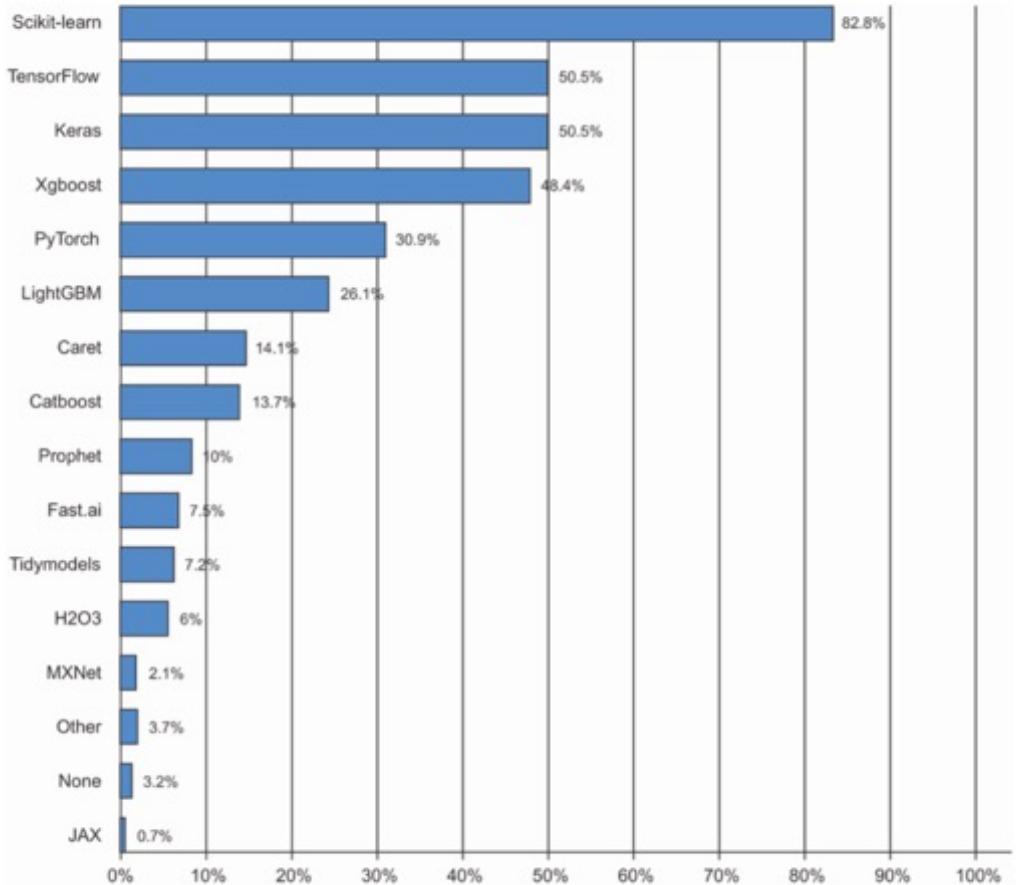


Figure 1.13 Tool usage across the machine learning and data science industry (Source: [www.kaggle.com/kaggle-survey-2020](http://www.kaggle.com/kaggle-survey-2020))



# Google Colab

The screenshot shows a Google Colab notebook titled "getting-started-keras.ipynb". The left sidebar contains a "Table of contents" with sections like "Getting started: Training and prediction with Keras in AI Platform", "Overview", "Dataset", "Objective", "Costs", "Before you begin", "Part 1. Quickstart for training in AI Platform", "Part 2. Quickstart for online predictions in AI Platform", and "Hyperparameter tuning". The main content area displays the "Getting started" section, which includes a heading, a "Run in Colab" button, and logos for Google Cloud, TensorFlow, and Keras. Below the heading, there are sections for "Overview", "Dataset", and a note about the dataset source.

getting-started-keras.ipynb

File Edit View Insert Runtime Tools Help

Table of contents

Getting started: Training and prediction with Keras in AI Platform

Overview

Dataset

Objective

Costs

Before you begin

Set up your local development environment

Set up your GCP project

Authenticate your GCP account

Create a Cloud Storage bucket

Part 1. Quickstart for training in AI Platform

Get training code and dependencies

Train your model locally

Train your model using AI Platform

Hyperparameter tuning

Part 2. Quickstart for online predictions in AI Platform

Getting started: Training and prediction with Keras in AI Platform

Run in Colab

View on GitHub

K + TensorFlow + Keras

Overview

This tutorial shows how to train a neural network on AI Platform using the Keras sequential API and how to serve predictions from that model. Keras is a high-level API for building and training deep learning models. [tf.keras](#) is TensorFlow's implementation of this API. The first two parts of the tutorial walk through training a model on Cloud AI Platform using prewritten Keras code, deploying the trained model to AI Platform, and serving online predictions from the deployed model. The last part of the tutorial digs into the training code used for this model and ensuring it's compatible with AI Platform. To learn more about building machine learning models in Keras more generally, read [TensorFlow's Keras tutorials](#).

Dataset

This tutorial uses the [United States Census Income Dataset](#) provided by the [UC Irvine Machine Learning Repository](#). This dataset contains information about people from a 1994 Census database, including age, education, marital status, occupation, and whether they make more than \$50,000 a year.

# Course Timeline

## AGENDA

- We will start with the concepts.
- We will then get into basic neural networks for simple prediction problems.
- Finally, we will get into more complex use cases, namely computer vision, NLP, and time series.

## NOTE TIMING OF DELIVERABLES

- First homework on basic prediction.
- Second homework on image-based prediction.
- There will be sign-ups for the week of 2/8 – 2/10, for your mid-point project check-in meeting.

## ONE FLEX WEEK

- This course is significantly revamped, so I reserve the right to revise our schedule depending on how we progress. Week 7 is tentatively going to cover autoencoders and perhaps another topic, but we may be playing catch-up at that point!

COURSE SCHEDULE  
(Subject to Revision Depending on Progress)

Week	Date	Topic	Assignments	Readings
1	1/20 (Th)	Course Logistics / Introduction	--	Chapter 1
2	1/25 (Tu) 1/27 (Th)	The Math Behind NNs Introduction to Keras and TensorFlow	-- --	Chapter 2 Chapter 3
3	2/1 (Tu) 2/3 (Th)	Neural Networks for Classification & Regression Model Tuning + Deep Dive on Keras	HW1 Posted Project Proposal Due --	Chapter 4 Chapters 5 & 7
4	2/8 (Tu) 2/10 (Th)	Mid-Point Project Check-in Meetings (via Zoom – Sign-up)	HW1 Due --	Chapter 6
5	2/15 (Tu) 2/17 (Th)	CNNs (Computer Vision)	HW2 Posted --	Chapter 8 Chapter 9
6	2/22 (Tu) 2/24 (Th)	RNNs (Time Series) RNNs (Text)	HW2 Due --	Chapter 10 Chapter 11
7	3/1 (Tu) 3/3 (Th)	Other Topics (e.g., Autoencoders, GANs)	-- In-Class Exam	
8			Spring Break	
9	3/15 (Tu) 3/17 (Th)	Final Project Presentations	-- Final Report Due	

# Course



## LECTURES

- We will meet twice weekly (beginning from Week 2), for ~2.5 hours each session. I will incorporate a 20-minute break in the middle of each session.
- The first half of each session will focus on lecture / concepts / explanation.

## HANDS-ON EXERCISES

- The second half of each session will be focused on implementing hands-on examples / demonstration in Python.
- I will provide you with Jupyter Notebooks and data-sets via GitHub, which we will walk through together.
- You are encouraged to ask questions as we progress.
- Note that the two homework assignments will be based on the in-class lab exercises / questions.

# Any Questions?

# What is ‘Deep’ (vs. Shallow) Learning?

# Where Deep Learning Started

Communicated by Dana Ballard

## Backpropagation Applied to Handwritten Zip Code Recognition

Y. LeCun  
B. Boser  
J. S. Denker  
D. Henderson  
R. E. Howard  
W. Hubbard  
L. D. Jackel

AT&T Bell Laboratories Holmdel, NJ 07733 USA

The ability of learning networks to generalize can be greatly enhanced by providing constraints from the task domain. This paper demonstrates how such constraints can be integrated into a backpropagation network through the architecture of the network. This approach has been successfully applied to the recognition of handwritten zip code digits provided by the U.S. Postal Service. A single network learns the entire recognition operation, going from the normalized image of the character to the final classification.

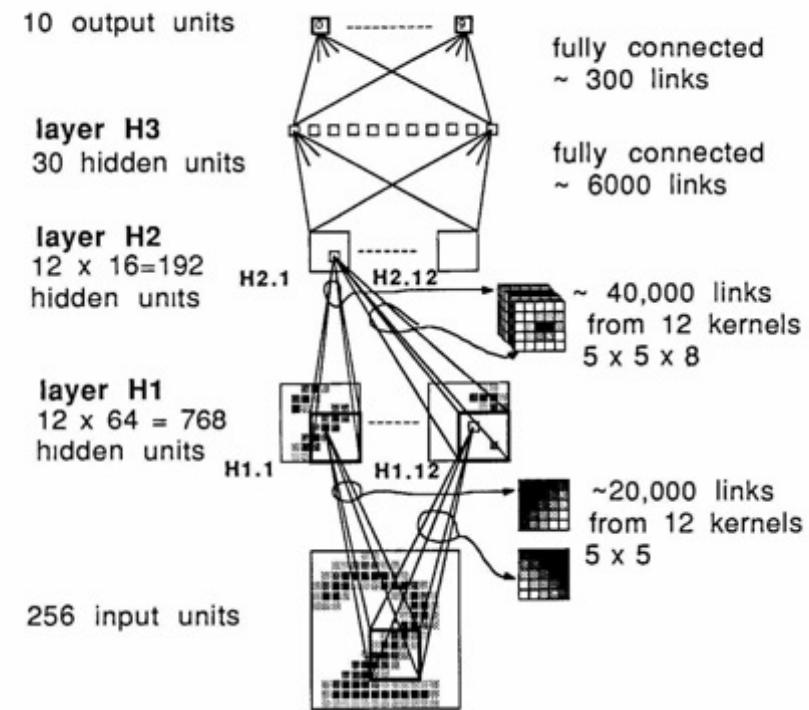


Figure 3 Log mean squared error (MSE) (top) and raw error rate (bottom) versus number of training passes

# Then It Shuffled Along for Decades...

## What was actually wrong with backpropagation in 1986?

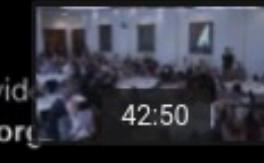
- We all drew the wrong conclusions about why it failed. The real reasons were:
  1. Our labeled datasets were thousands of times too small.
  2. Our computers were millions of times too slow.
  3. We initialized the weights in a stupid way.
  4. We used the wrong type of non-linearity.

A few years ago, Jeff Dean decided that with enough computation, neural networks might do amazing things.

He built a lot of infrastructure to allow big neural nets to be trained on lots of cores in Google data centers.

THE  
ROYAL  
SOCIETY





Watch more vid  
[royalsociety.org](http://royalsociety.org) 42:50

# Now...

ARTIFICIAL INTELLIGENCE

# A GPT-3 bot posted comments on Reddit for a week and no one noticed

Under the username /u/thegentlemetre, the bot was interacting with people on /r/AskReddit, a popular forum for general chat with 30 million users.

By Will Douglas Heaven

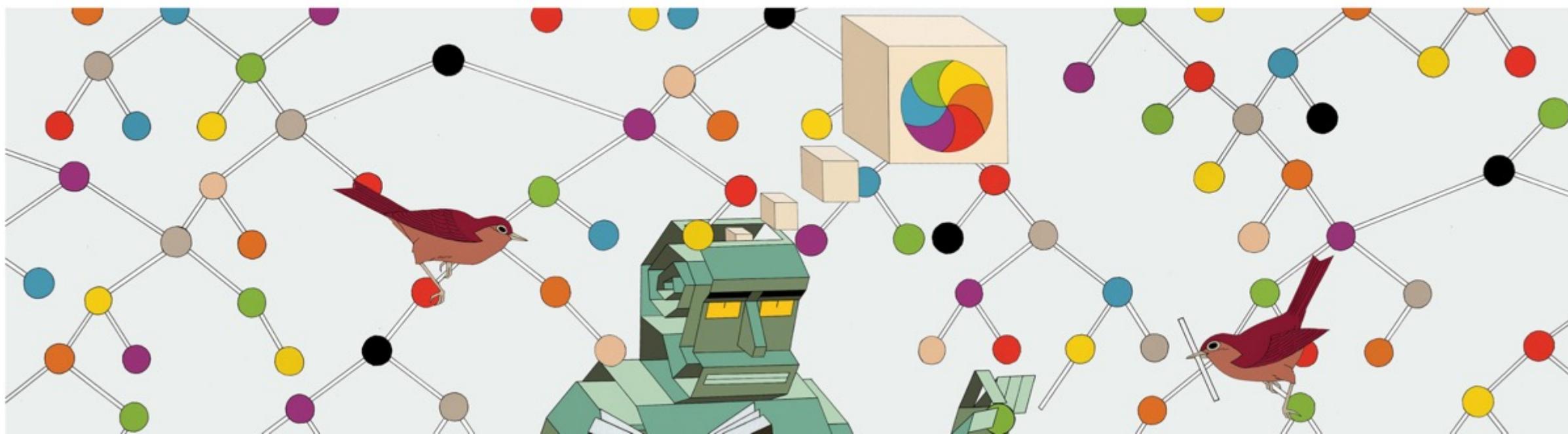
October 8, 2020

ARTIFICIAL INTELLIGENCE

# Symbolic Mathematics Finally Yields to Neural Networks

35 |

*After translating some of math's complicated equations, researchers have created an AI system that they hope will answer even bigger questions.*



---

# Predicting Parameters in Deep Learning

---

**Misha Denil<sup>1</sup> Babak Shakibi<sup>2</sup> Laurent Dinh<sup>3</sup>**

**Marc'Aurelio Ranzato<sup>4</sup> Nando de Freitas<sup>1,2</sup>**

<sup>1</sup>University of Oxford, United Kingdom

<sup>2</sup>University of British Columbia, Canada

<sup>3</sup>Université de Montréal, Canada

<sup>4</sup>Facebook Inc., USA

{misha.denil,nando.de.freitas}@cs.ox.ac.uk

laurent.dinh@umontreal.ca

ranzato@fb.com

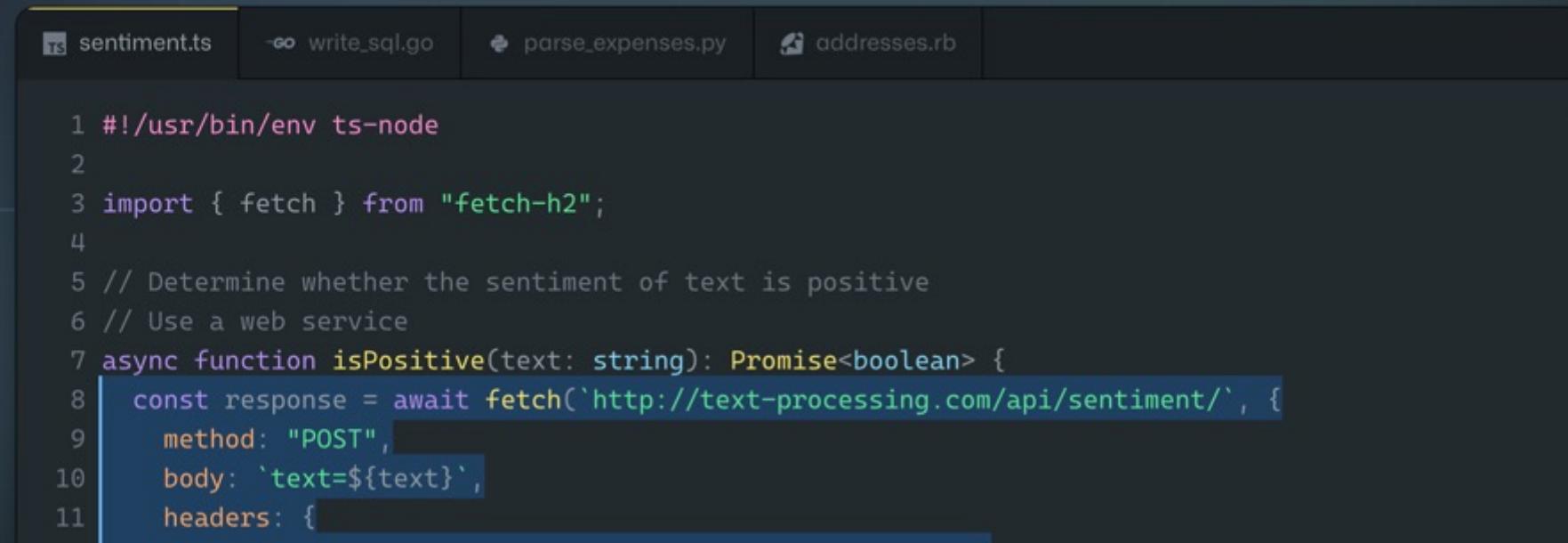
## Abstract

We demonstrate that there is significant redundancy in the parameterization of several deep learning models. Given only a few weight values for each feature it is possible to accurately predict the remaining values. Moreover, we show that not only can the parameter values be predicted, but many of them need not be learned at all. We train several different architectures by learning only a small number of weights and predicting the rest. In the best case we are able to predict more than 95% of the weights of a network without any drop in accuracy.

Technical Preview

# Your AI pair programmer

With GitHub Copilot, get suggestions for whole lines or entire functions right inside your editor.

[Sign up >](#)

A screenshot of a dark-themed code editor interface. At the top, there are four tabs: 'sentiment.ts' (selected), '-go write\_sql.go', 'parse\_expenses.py', and 'addresses.rb'. The main editor area displays the following TypeScript code:

```
1 #!/usr/bin/env ts-node
2
3 import { fetch } from "fetch-h2";
4
5 // Determine whether the sentiment of text is positive
6 // Use a web service
7 async function isPositive(text: string): Promise<boolean> {
8   const response = await fetch(`http://text-processing.com/api/sentiment/`, {
9     method: "POST",
10    body: `text=${text}`,
11    headers: {
```

The line 'const response = await fetch(`http://text-processing.com/api/sentiment/`, {' is highlighted with a blue selection bar, indicating it is the current suggestion being previewed.

Money Stuff

# Sorry, Zillow's Computer Can't Buy Your House Right Now

Also CEO pay, the Boredom Markets Hypothesis and Big Short guys being big short.

By [Matt Levine](#) [+Sign Up](#)

October 18, 2021, 1:18 PM EDT

---

Matt Levine is a Bloomberg Opinion columnist covering finance. He was an editor of Dealbreaker, an investment banker at Goldman Sachs, a mergers and acquisitions lawyer at Wachtell, Lipton, Rosen & Katz, and a clerk for the U.S. Court of Appeals for the 3rd Circuit.

[Read more opinion](#)

## Zillow

Deciding how much you should pay for a share of large-cap publicly traded stock is not an *entirely* solved problem, but it's pretty close. If someone comes to you and says "hey I have 100 shares of Microsoft Corp. stock for sale, how much will you pay me for it," a pretty decent answer would be to look at the last price at which Microsoft traded – like a millisecond ago – and subtract, you know, one cent from that price. That will get you a price that is likely to be competitive (the seller might actually sell to you), likely to be profitable (you might be able to sell it for more than you paid), and



EDITORS' PICK | Oct 14, 2021, 07:01am EDT | 79,274 views

# Fraudsters Cloned Company Director's Voice In \$35 Million Bank Heist, Police Find



Thomas Brewster Forbes Staff

Cybersecurity

Associate editor at Forbes, covering cybercrime, privacy, security and surveillance.

f t in





Gord Burtch  
@gburtch

...

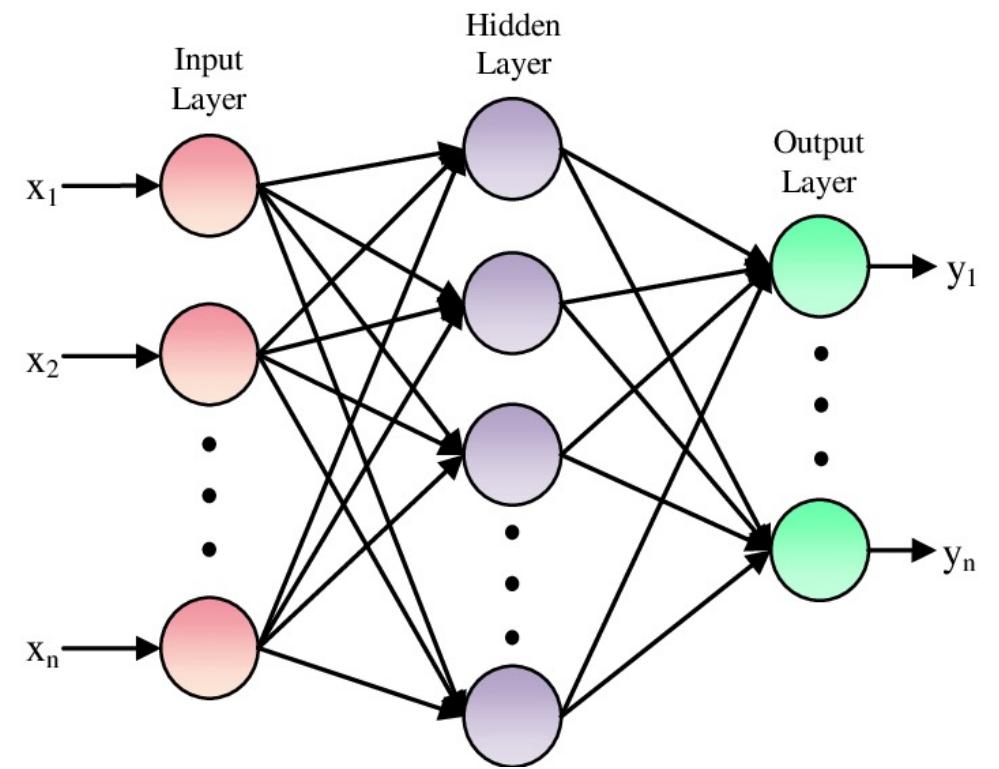
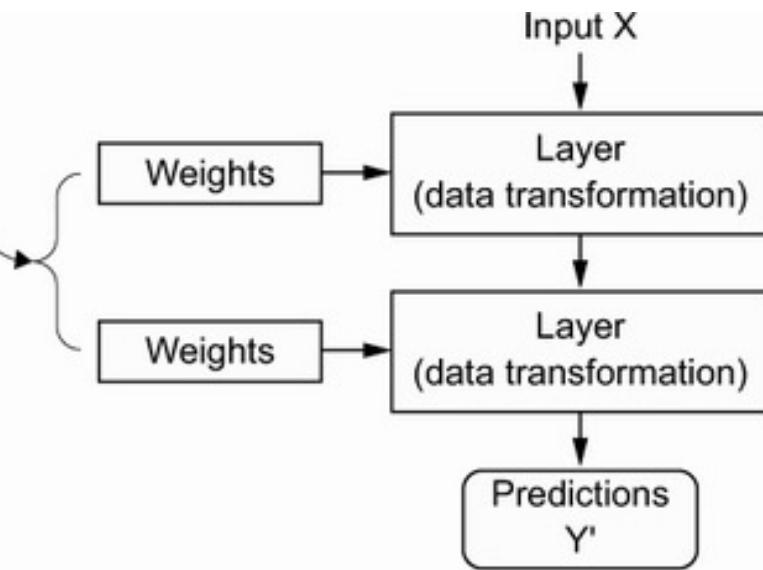
I made this video for free in 5-10 minutes, using my phone, a couple git repositories, ffmpeg and Google colab. The image is based on one picture of trump, and 5 seconds of audio from a speech. Just imagine what you someone can do with time, effort and skill...



# How It Works, Conceptually

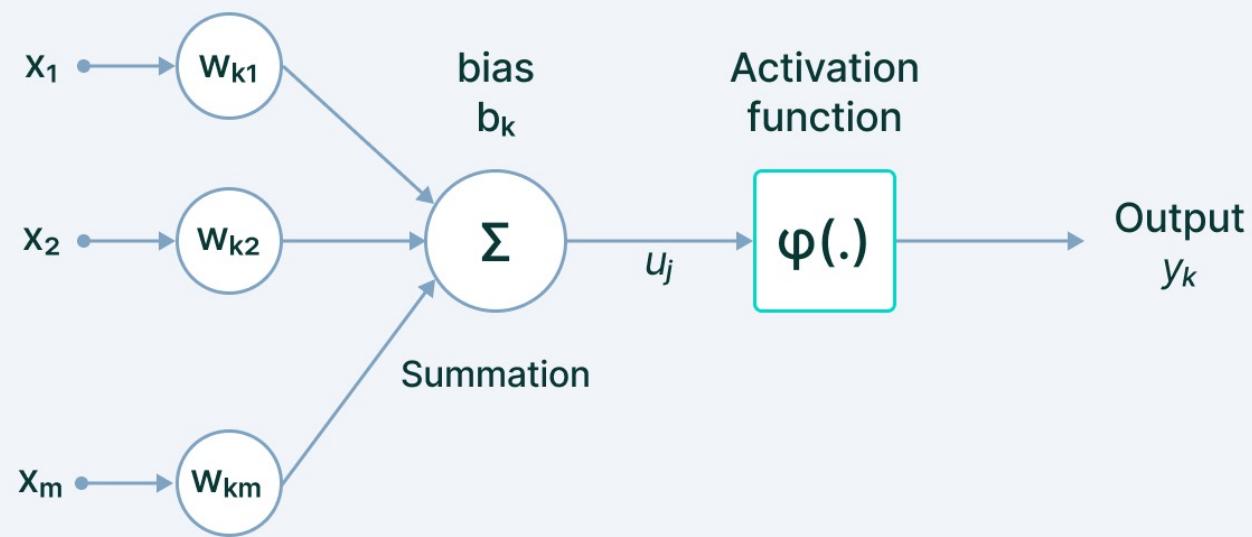
# Model Parameters

**Goal: finding the right values for these weights**

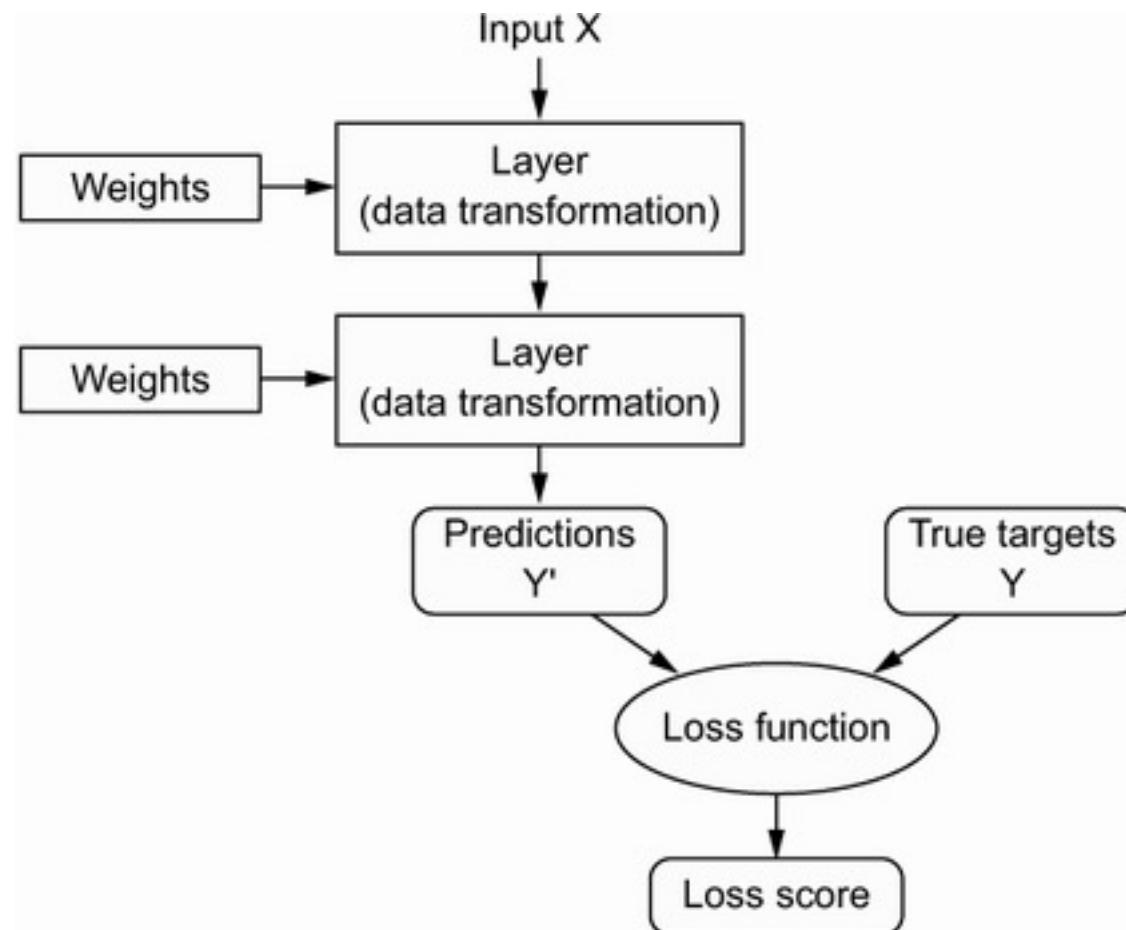


# Model Parameters

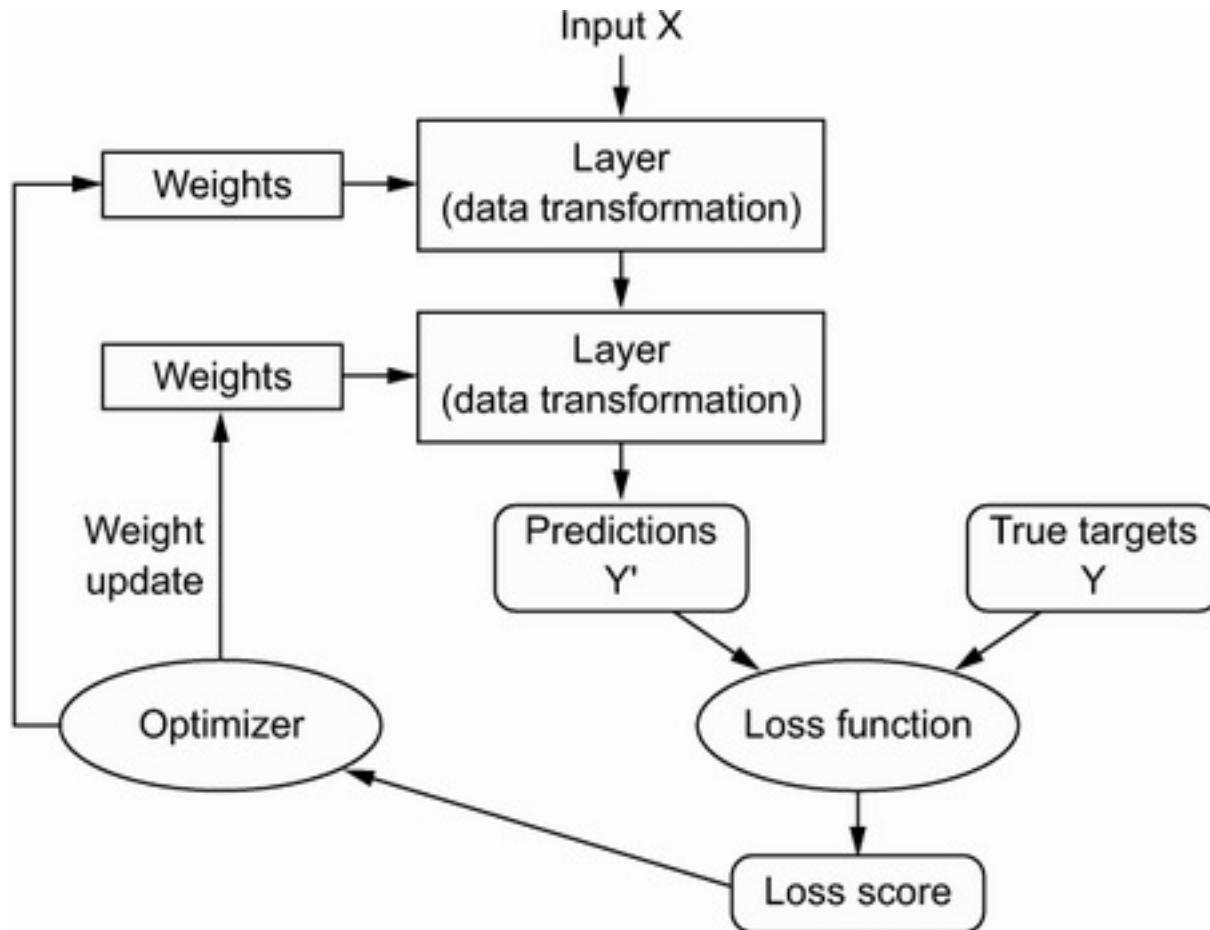
## Neuron



# Loss Function (Error)



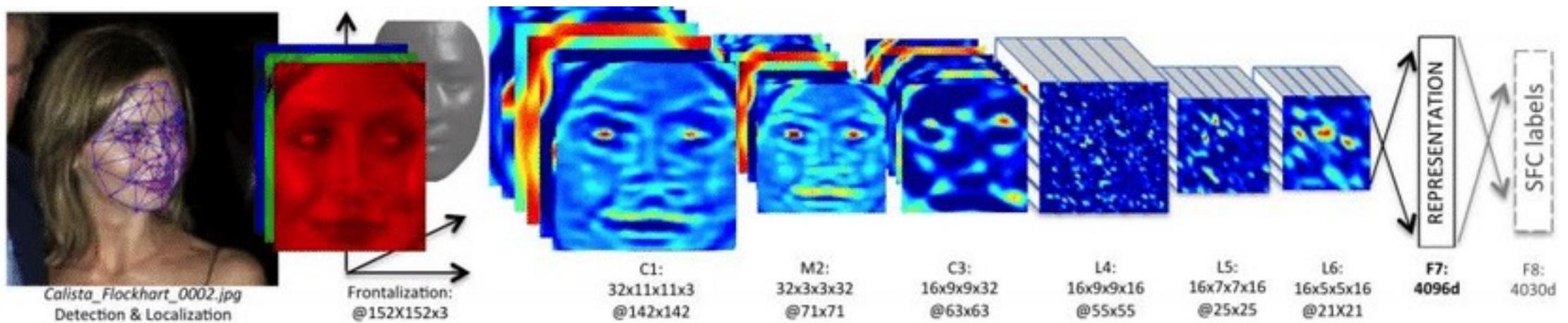
# Optimization



# When to Learn Deeply (vs. Not)

## COMPLEX RELATIONSHIPS

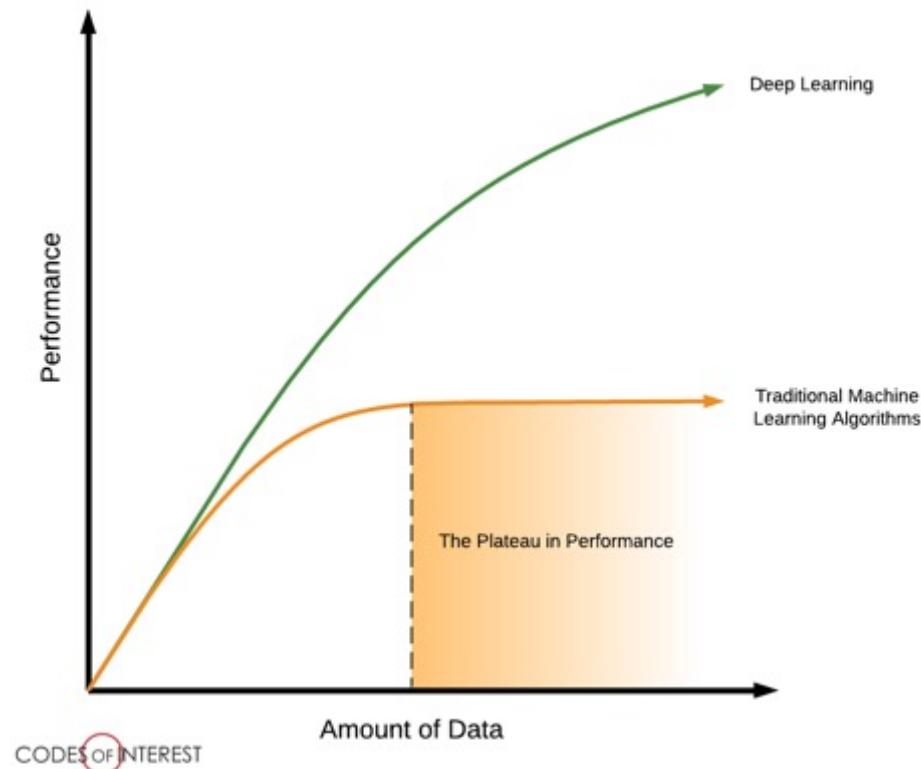
- Complex, non-linear, interactive relationships and mappings; common use cases involve unstructured (high dimensional) data. Deep learning techniques remove the need for feature engineering, a daunting task.



# When to Learn Deeply (vs. Not)

## LOTS OF DATA ON HAND

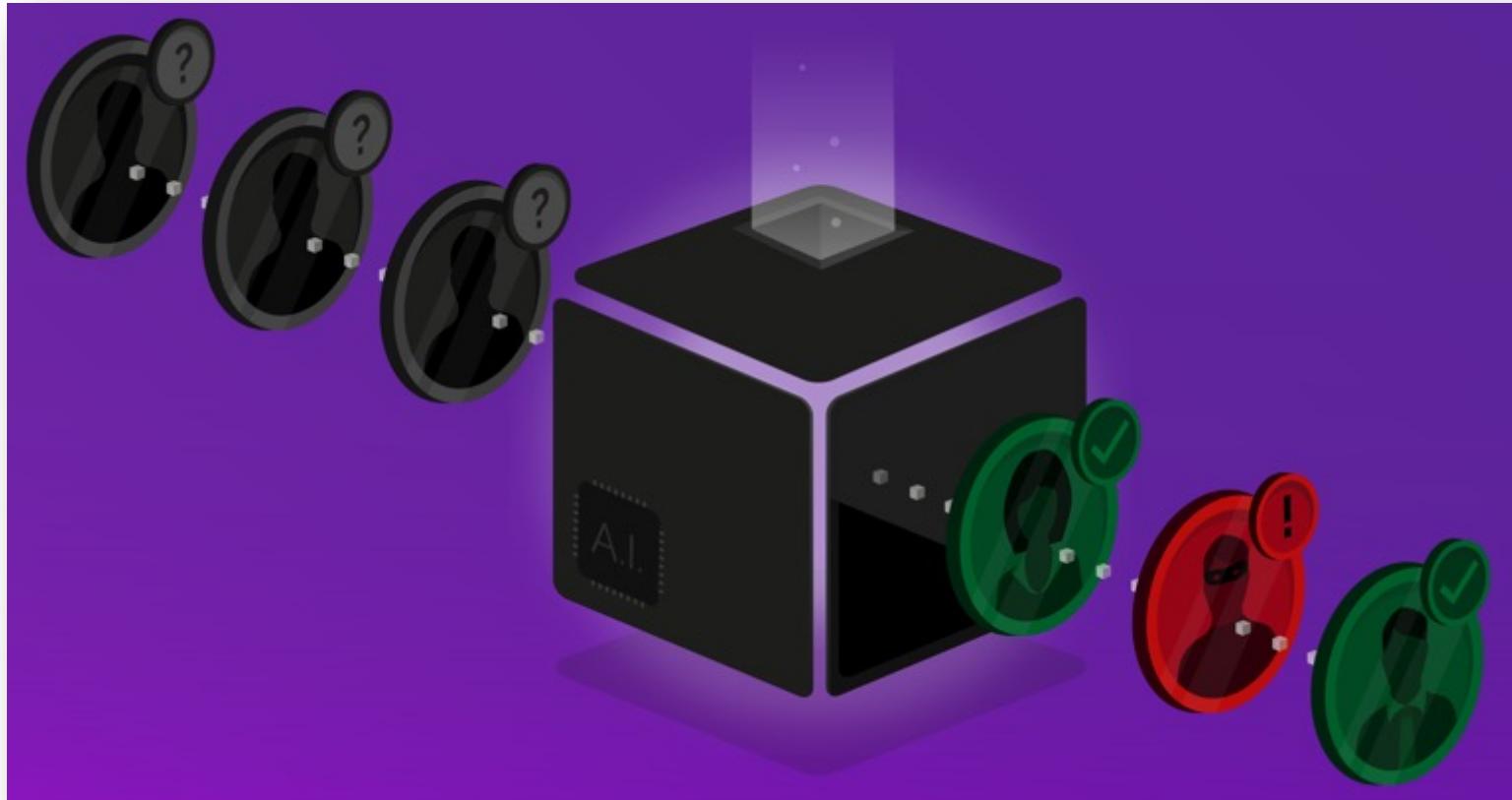
- To be able to learn those complex mappings, typically requires many, many, many training examples.



# When to Learn Deeply (vs. Not)

## LITTLE NEED FOR UNDERSTANDING

- Although there have been advancements in explainable and interpretable AI, deep nets are notoriously “black box” algorithms.

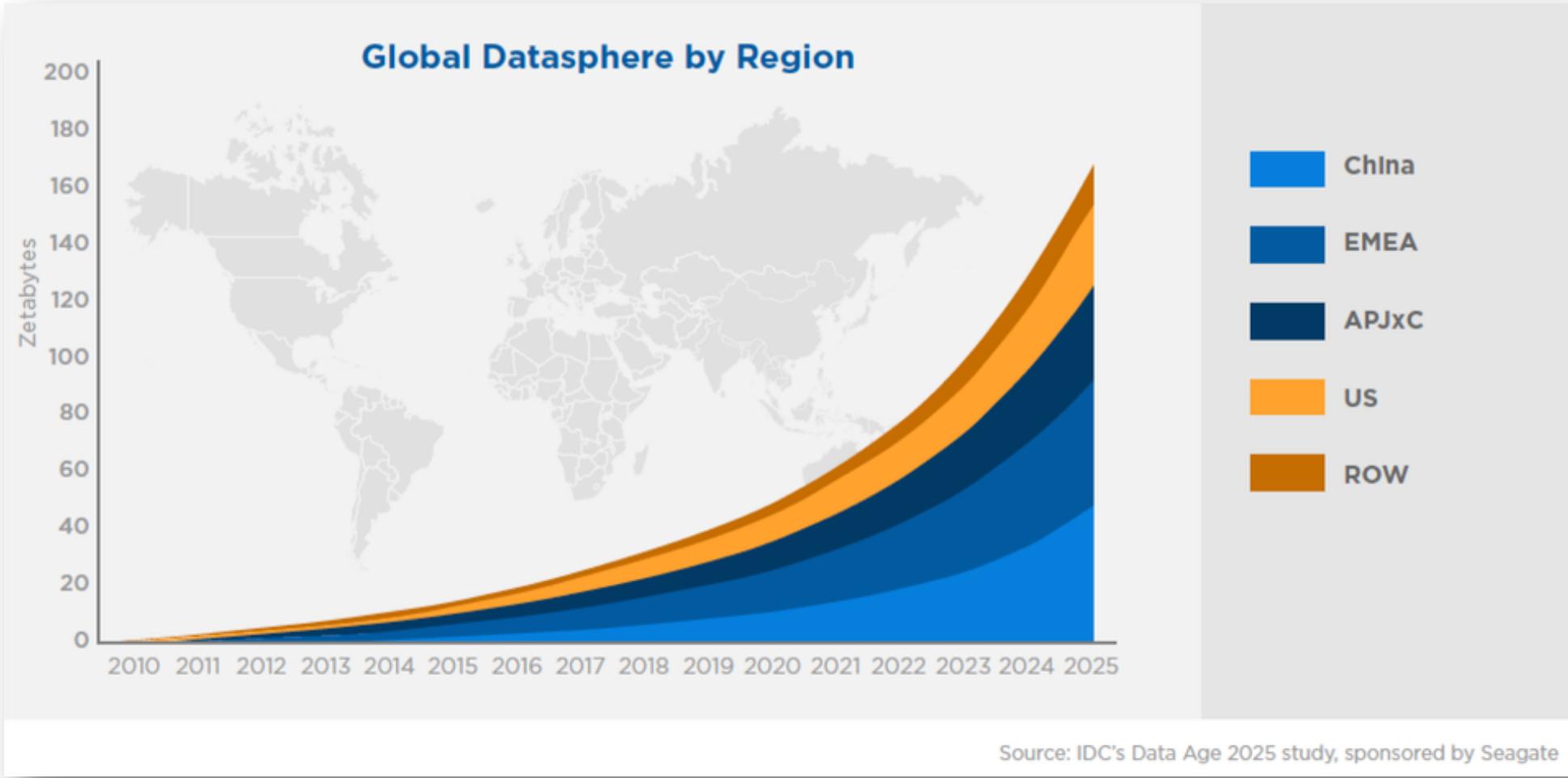


# Why Did Deep Learning Take Off?

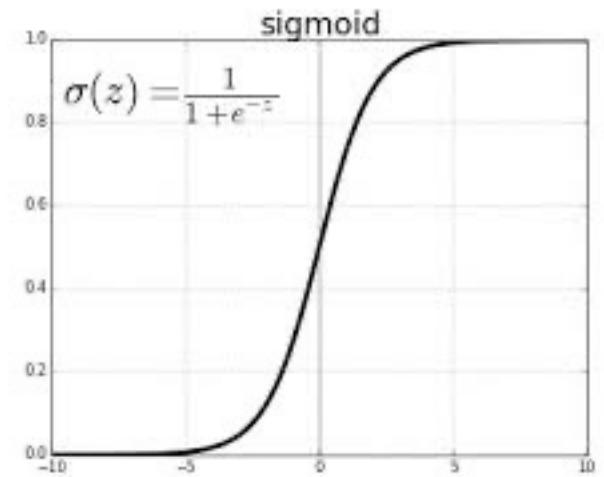
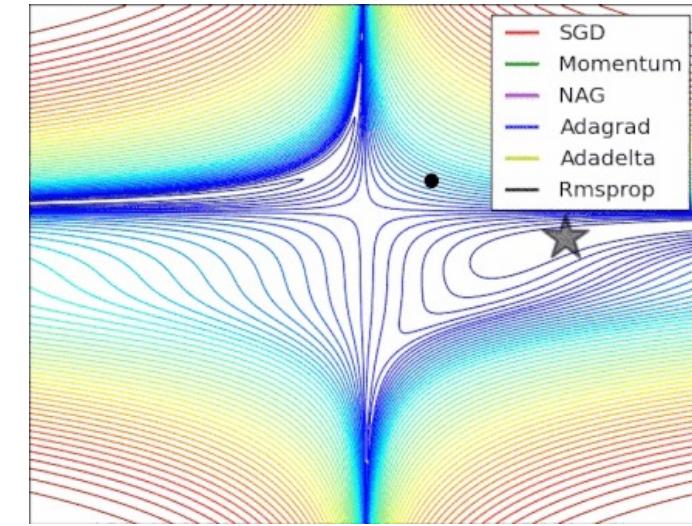
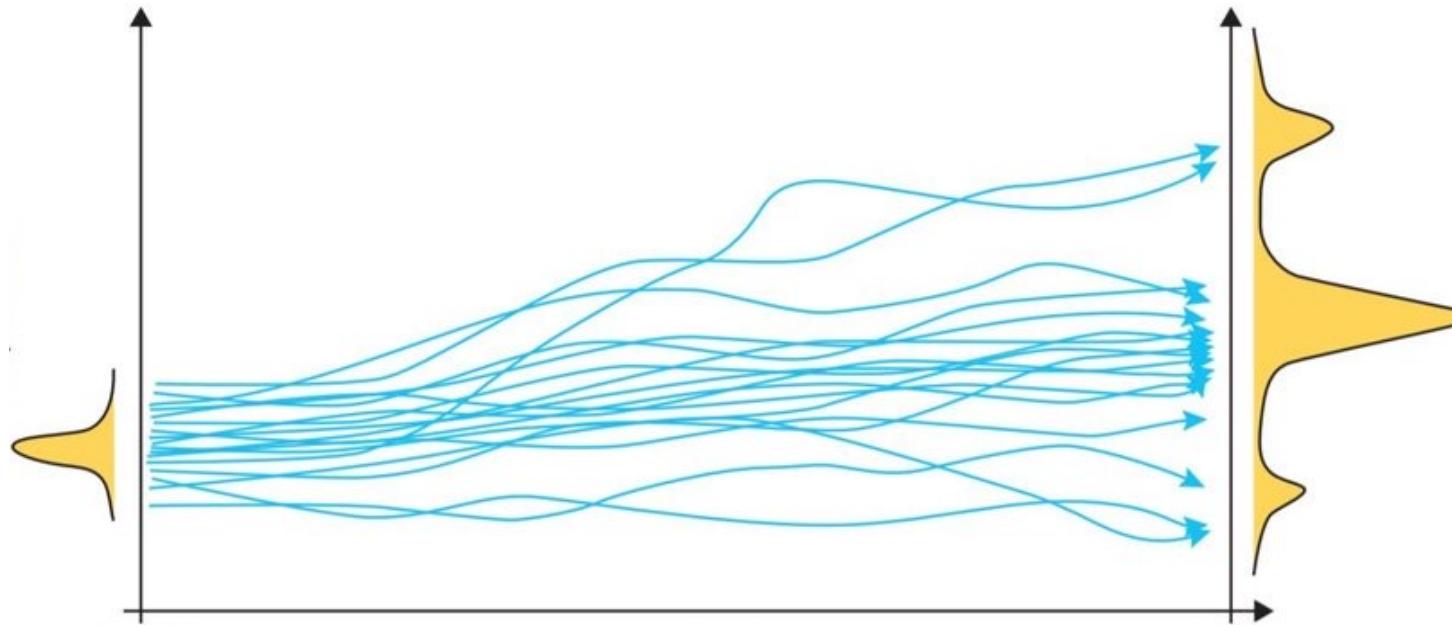


# Video Games

# Data



# Algorithmic Improvements



# Questions?

# Walk-Through of Google Colab

