The Evolution Overview - Five Phases

The Journey from Theory to Generative AI

1. **Teaching Machines to Calculate (1940s-1970s)**
2. **Expert Systems and Learning Algorithms (1980s-2000s)**
3. **Big Data and Distributed Computing (2000s-2010s)**
4. **Deep Learning Revolution (2010-2017)**
5. **Large Language Models and Generative AI (2017-Present)**

Each phase built essential components that made today's AI possible

Phase 1 - Teaching Machines to Calculate (1940s-1970s)

**Foundation Period: Mathematical and computational frameworks

Key Contributors:

- **Ronald Fisher:** Maximum likelihood estimation, ANOVA, experimental design
- **Neyman-Pearson:** Hypothesis testing framework (foundation for A/B testing)
- **Claude Shannon:** Information theory, entropy concept

ENIAC's programmers at work. On the left is Betty Jean Jennings, and Fran Bilas is on the right.

Photo courtesy the U.S. Army and the archives of the ARL Technical Library

Photo courtesy of the U.S. Army

The ENIAC (Electronic Numerical Integrator and Computer) was the first fully electronic digital computer, designed by John W. Mauchly and J. Presper Eckert Jr. at the University of Pennsylvania. Unveiled in 1946, it was created for the U.S. Army to rapidly calculate artillery trajectories during World War II. This revolutionary machine featured 18,000 vacuum tubes, weighed over 30 tons, and occupied nearly 2,000 square feet.

Quality Control Example - Statistical Testing

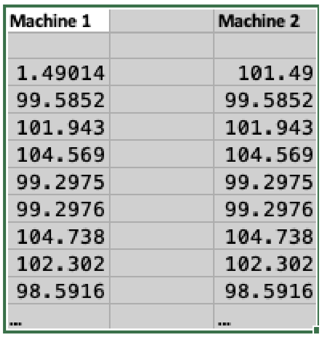
Factory Scenario:

- Standard weight: 100 grams
- Natural variation expected
- Need to detect when machines malfunction

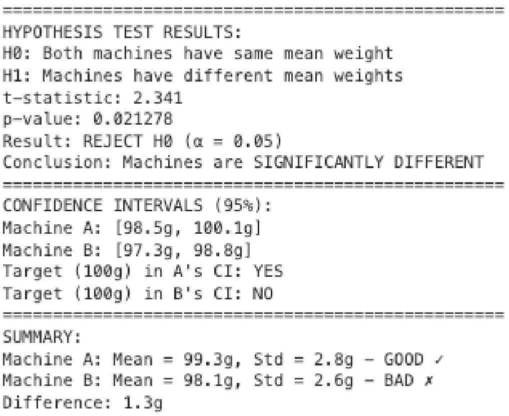
Statistical Framework:

- **Null Hypothesis:** Machine produces correct weight
- **Alternative:** Machine is off-target
- **Decision Rule:** If sample mean differs significantly, investigate

Business Impact: Prevent defective products, maintain quality standards



Machine 1		Machine 2
1.49014		101.49
99.5852		99.5852
101.943		101.943
104.569		104.569
99.2975		99.2975
99.2976		99.2976
104.738		104.738
102.302		102.302
98.5916		98.5916
...		...



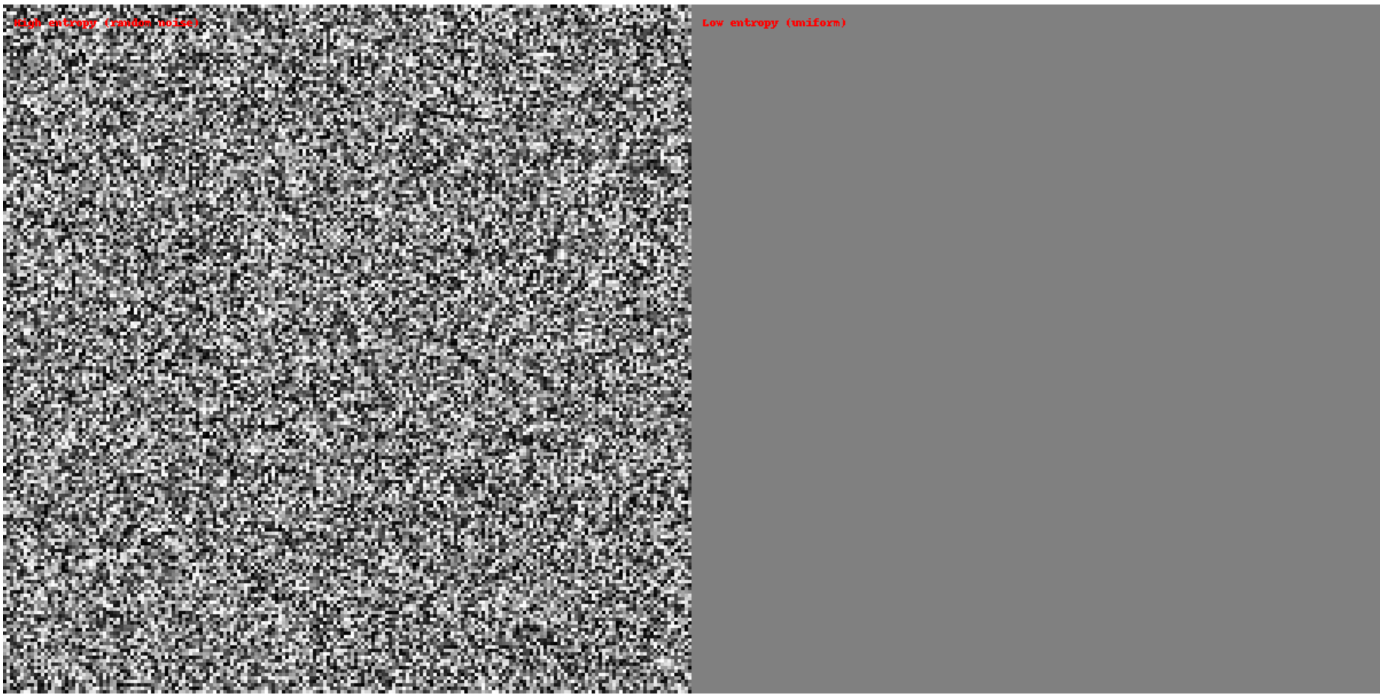
```
==================================================
HYPOTHESIS TEST RESULTS:
H0: Both machines have same mean weight
H1: Machines have different mean weights
t-statistic: 2.341
p-value: 0.021278
Result: REJECT H0 (α = 0.05)
Conclusion: Machines are SIGNIFICANTLY DIFFERENT
==================================================
CONFIDENCE INTERVALS (95%):
Machine A: [98.5g, 100.1g]
Machine B: [97.3g, 98.8g]
Target (100g) in A's CI: YES
Target (100g) in B's CI: NO
==================================================
SUMMARY:
Machine A: Mean = 99.3g, Std = 2.8g - GOOD ✓
Machine B: Mean = 98.1g, Std = 2.6g - BAD ✗
Difference: 1.3g
```

Information Theory and Entropy

** Shannon's breakthrough (1948) boils down to the fact that a surprising messages contain more information than predictable ones

- Entropy: Average amount of surprise/uncertainty in a message
- Examples:
  - High entropy: Random password (unpredictable)
  - Low entropy: String of repeated letters (predictable)
- Modern Applications:
  - File compression, data transmission
  - How AI systems learn patterns
  - Foundation for machine learning optimization (including for LLMs)

Information Theory and Entropy - Cont'd



Perplexity: Measuring AI Language Quality

What is Perplexity? How surprised the AI is by the next word in a sentence

Examples:

- **Low perplexity (good)**: "The sun rises in the..." → AI expects "east"
- **High perplexity (bad)**: "The purple elephant quantum..." → AI has no clue

Mathematical relationship: $Perplexity = 2^(Cross-entropy loss)$

Intuitive Understanding:

- Perplexity = 8 means AI is as confused as choosing randomly between 8 options
- Good models: 10-30 on news text
- Bad models: 100+

Perplexity is one of the metrics used for comparing major language models (GPT, Claude, etc.)

Perplexity Examples - 1

1. CONFIDENT AI - Predicting next word after 'Happy Birthday to...' AI's prediction: 'you' (95% confident) Entropy: 0.36 bits Perplexity: 1.29 (feels like choosing between ~1 option)

2. UNCERTAIN AI - Predicting next word after 'The weather is...' AI is unsure between multiple options Entropy: 2.42 bits Perplexity: 5.36 (feels like choosing between ~5 options)

3. CONFUSED AI - Predicting next word after 'Xqwz vplm qrst...' AI has no idea what comes next (gibberish input) Entropy: 6.84 bits Perplexity: 114.82 (feels like choosing between ~115 options!)

The Birth of Neural Networks

McCulloch-Pitts Model (1943): The First "Brain Cell" in Math

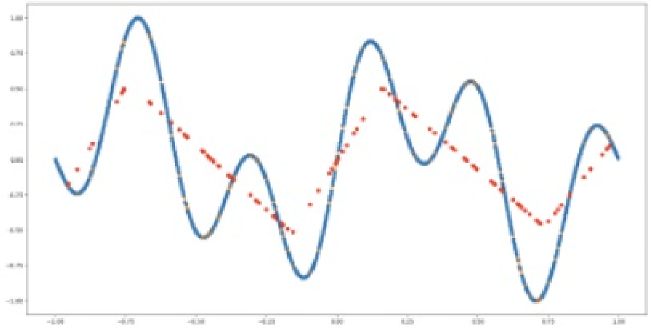
- Binary neurons that either "fires" (1) or "doesn't fire" (0) - like an on/off switch
- Revolutionary idea that showed that networks of these simple neurons (switches) can perform ANY logical computation
- Think of it like as a digital displays using thousands of on/off pixels to show smooth images

Hebb's Rule (1949): How Brains Learn

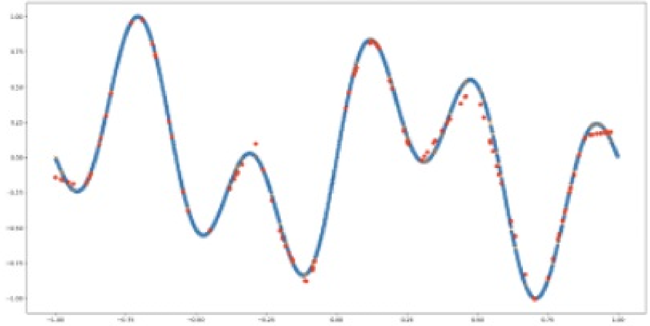
- Proposed that "Neurons that fire together, wire together"
  - In practice, if two brain cells activate simultaneously, their connection strengthens

- Modern AI, this became "backpropagation" in deep learning, which is used to "calibrate"

- These 1940s insights launched the entire field of artificial intelligence.



Data approximation (red) using model with 62 parameters



Data approximation (red) using model with 4,491 parameters

https://blog.cubieserver.de/2019/approximate-function-with-neural-network/

Phase 2 - Expert Systems and Learning (1980s-2000s)

Two Major Developments:

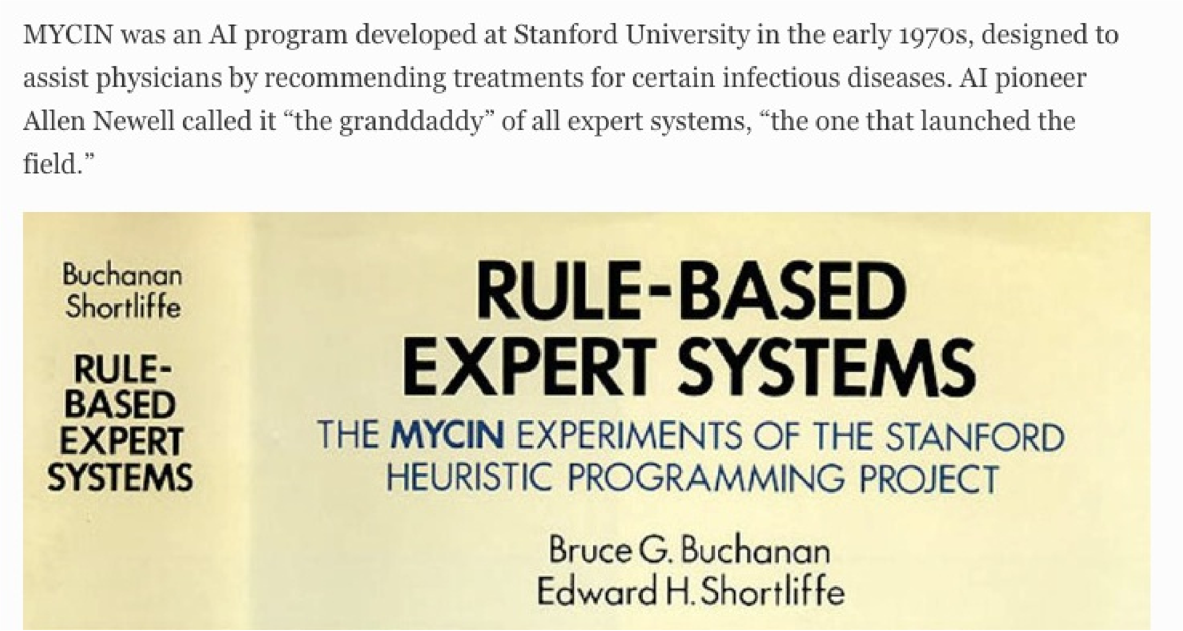
1. Expert Systems - Machines That Follow Rules

- Encoded human expertise as IF-THEN rules
- MYCIN: 69% accuracy in medical diagnosis
- Matched human expert performance

2. Learning Algorithms - Machines That Find Patterns

- Support Vector Machines (SVMs)
- Ensemble methods (Random Forests, AdaBoost)
- Shift from symbolic to statistical learning

MYCIN Infectious Diseases treatment Recommnedation System



MYCIN was an AI program developed at Stanford University in the early 1970s, designed to assist physicians by recommending treatments for certain infectious diseases. AI pioneer Allen Newell called it “the granddaddy” of all expert systems, “the one that launched the field.”

Buchanan
Shortliffe
RULE-BASED EXPERT SYSTEMS

RULE-BASED EXPERT SYSTEMS
THE MYCIN EXPERIMENTS OF THE STANFORD HEURISTIC PROGRAMMING PROJECT
Bruce G. Buchanan
Edward H. Shortliffe

Forbes Article

How Expert Systems Worked

Step-by-Step Process:

1. Q: "Does the patient have a fever?" A: Yes
2. Q: "Is the fever above 38.5°C?" A: Yes
3. Q: "White blood cell count > 12,000?" A: Yes

Rule Chaining (simplified):

- Rule 1: IF fever > 38.5°C AND WBC > 12,000 THEN bacterial infection (0.8 confidence)
- Rule 2: IF bacterial infection AND hospitalized THEN antibiotics (0.9 confidence)

Innovation: Separation of knowledge base (rules) from inference engine (logic)

Expert Systems - Limitations

- These expert system had to two major issues
- Brittleness:
- Failed completely with unexpected inputs
  - Ask about unlisted disease yeilds no answer
- Thousands of interdependent rules
  - One change could break entire system
- Knowledge Acquisition Bottleneck:
  - Extracting rules from human experts took years
    - Imagine documenting every decision rule doctors use

The Statistical Learning Revolution

- Major Shift: From symbolic to statistical learning

- Support Vector Machines (SVMs) - Cortes & Vapnik (1995):

- Built on theory by Vapnik & Chervonenkis (1960s-70s)
- Found optimal boundaries between data categories
- Were widely used for spam vs ham email classification in early 2000s
  - SVM finds the widest possible bopundary separating spam from legitimate emails

- Key Insight: Let machines learn patterns from data rather than programming explicit rules

- Why this mattered:

  - No need to manually write "IF email contains 'FREE MONEY' THEN spam"
  - System learns from thousands of examples what patterns indicate spam
    - Could handle complex combinations that are complex to describe and that humans might miss

Ensemble Methods - Wisdom of Crowds

Random Forests (1996):

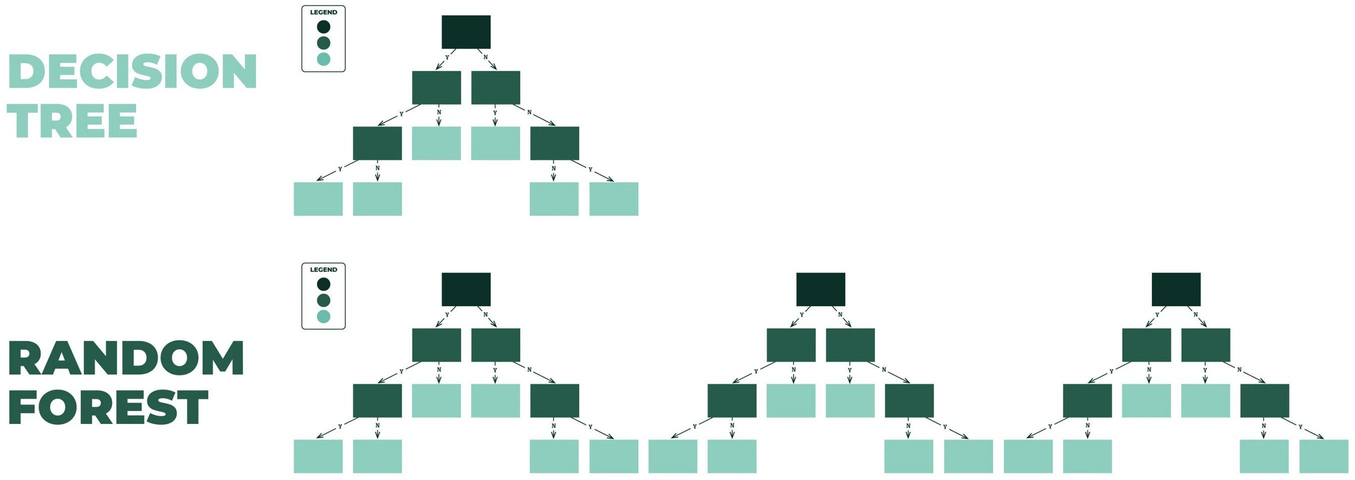
- Create hundreds of different model using subsets of data
  - Widely used with decision trees
- Average their "votes" for final answer
  - Like asking 100 doctors for opinions on subset of the data and then taking majority

Ensemble Methods - Wisdom of Crowds

- Random Forests (Breiman, 2001):

- Create hundreds of different models using subsets of data
- Widely used with decision trees
- Average their "votes" for final answer
- Like asking 100 doctors for opinions on subset of the data and then taking majority

- The key insight is that multiple "weak" models together often outperform single "strong" model



Backpropagation Breakthrough (1986)

Algorithm proposed by Rumelhart, Hinton, and Williams to train neural networks

- Method for models to learn from mistakes
    - Made multi-layer neural networks trainable
- How It Works:

1. Calculate prediction error
2. Adjust network's internal numbers to reduce error
3. Repeat thousands of times
4. Network gradually learns patterns

- Enabled the deep learning revolution that followed decades later

Phase 3 - Big Data and Distributed Computing (2000s-2010s)

- The Data Explosion:
    - 2010: Humanity generated 5 exabytes every 2 days
    - Equivalent to all recorded human words throughout history
    - New reality: Data generation >> Processing capacity
- The Challenge: is not storage, but computation
    - Single computer would take 11 days to read 100 terabytes
    - Yet, we need answers across billions of records in seconds

MapReduce - Paradigm Shift (2004)

- Google's Innovation: Smarter work distribution, not faster hardware

- Two-Phase Framework:

- **Map:** Process data in parallel across hundreds or thousands of computers

- **Reduce:** Aggregate results

- Google's web index: 1 computer = 4 years

- 1,000 computers with MapReduce = few hours

- MapReduce rendered many business questions are "embarrassingly parallel"

Hadoop - Democratizing Distributed Computing

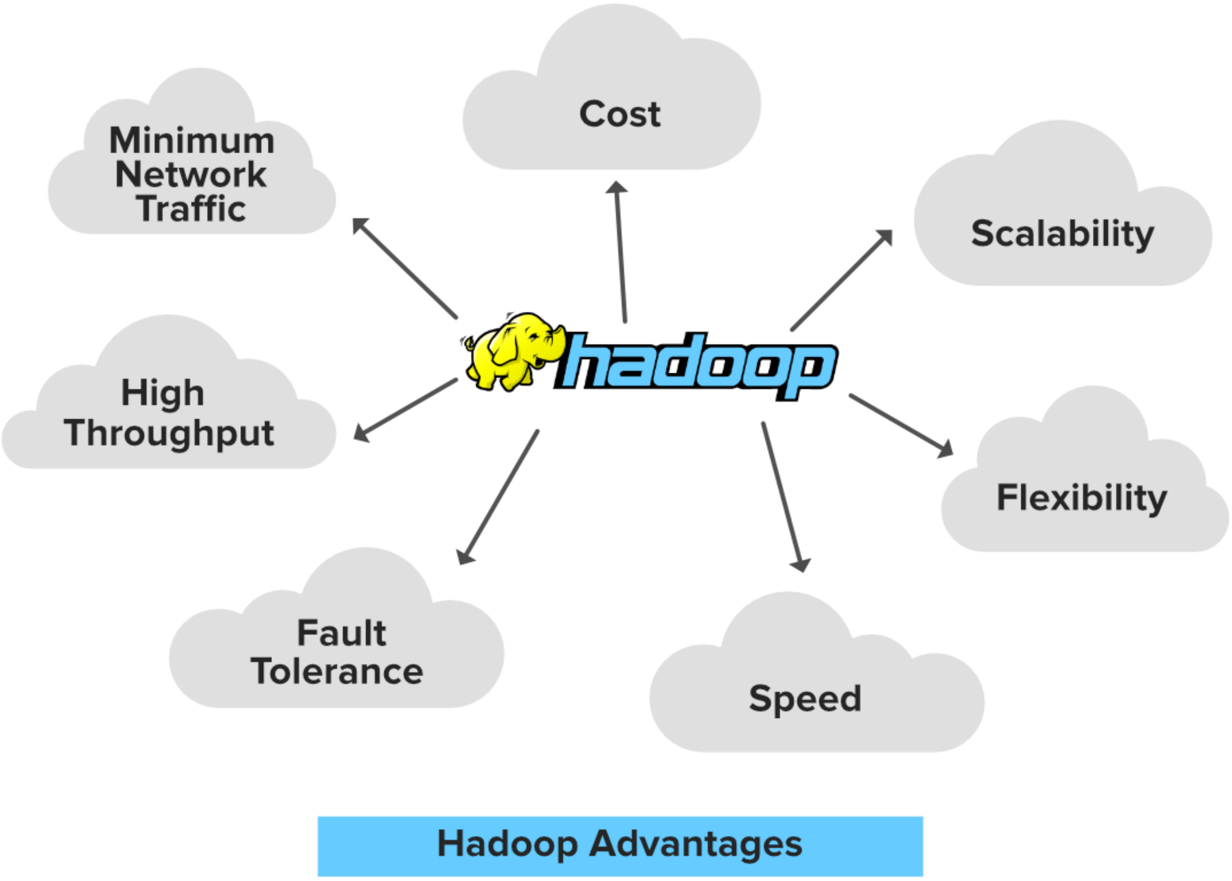
- Before Hadoop:

- Only tech giants could afford distributed computing
- Expensive specialized hardware required and complex configuration and administration

- Hadoop's proposed a framework to run MapReduce in commodity hardware (same servers for websites)

- Start with a few machines, scale to thousands
- Yahoo 2008: 10TB daily on 910 machines at 1/10th traditional cost

- Now small companies could access enterprise-scale computing



Spark - Solving the Speed Problem

- MapReduce Limitation:
    - Wrote intermediate results to disk
    - Like saving/reloading document after every edit
    - Machine learning needed hundreds of data passes
- Spark's Solution:
    - Kept data in RAM
    - 100x speedups for machine learning workloads
    - Enabled real-time analytics
- Business Example: Uber optimizing driver placement every few seconds based on current demand

NoSQL - Why Traditional Databases Couldn't Handle the Internet

- Traditional databases (SQL) were strict: document must have exactly the same fields.
    - Example: Every "receipt" record needs: Vendor, Adress, Phone, items,
        - Some receipts don't have a physical address, some have discount fieds, some have barcodes, etc.
- NoSQL provides flexible structure through 3 main approaches

1. Document stores (like MongoDB): Store entire "documents" (those are python dictionary-like objects) with their own structure.
    - dictionnary fields don't matter at storage time.
2. Graph databases : Store relationships directly
3. Key-value stores: Lightning-fast simple lookups

Cloud Computing Economics

- The pay-per-use model was a game changer and still the backbone for genAI innovation

- Example Scenario:

- Pharmaceutical company needs drug discovery simulations
- Rent 10,000 computers for hours and pay only for actual usage

- Impact:

- Eliminated capital expenditure barrier
- Small teams can have Supercomputer-scale resources
  - Netflix 2010: 1 billion viewing hours monthly, zero owned servers

The Datafication Phenomenon

- Everything decame a source for data
  - Phone sensors: 40 data points per second
  - Credit card transactions: Economic patterns
  - Social media: Cultural trends
- Beyond Digitization:
  - Transformation of human behavior into mathematical patterns
  - Algorithms could learn from daily activities
- Lead to new capabilities, from real-time personalization and predictive analytics to behavioral insights.

Phase 4 - Deep Learning Revolution (2010-2017)

- Three Forces Converged:

1. Massive datasets from Big Data era
2. Graphics cards accelerated training 50x faster
   - NVIDIA CUDA originally for game developers
   - Millions of simple calculations in parallel
   - 3 months CPU training can be computed in less than 3 days on GPU
3. Mathematical breakthroughs made deep networks trainable and more efficient

Framework Democratization

- Before 2015:
  - Implementing neural networks required months of specialized programming
  - Limited to expert researchers with deep technical knowledge
- TensorFlow (2015) & PyTorch (2016):
  - Reduced implementation time to days or hours
  - Like difference between building car from scratch vs. assembling components
  - Graduate students could implement state-of-the-art AI
- Result: Tens of thousands of developers by 2017; explosion of AI applications.

Training Deep Networks - Technical Breakthroughs

- The Problem:
  - Deeper networks promised better performance (theorerically) but were nearly impossible to train
  - Signals got lost through dozens of layers
  - Like telephone game going wrong
- New developments

- Better optimizers: Adapted learning rates automatically
- Smart cruise control** that adjusts based on conditions
  - Enabled growth ~60 million parameters in AlexNet and ResNet to trillons of parameters (GPT-4)

- This unlocked the full potential of neural network depth

The Transformer Architecture (2017)

- "Attention Is All You Need" Paper was a fundamental breakthrough

- Previously, neural networks read text sequentially

- Like looking at one word at a time
- Struggled to remember distant context

- Transformer solution proposed Self-attention mechanism

- See all words simultaneously
- Understand how they relate to each other
- Parallel processing instead of sequential

Example: "He sat by the bank of the river"

- Attention helps recognize "bank" means shore, not financial institution
- Does this by "attending to" the word "river"

- Future Impact: Foundation for ChatGPT and modern language models

Scale Reveals Unexpected Intelligence

- GPT-3 (2020): 175 Billion Parameters

- 100x larger than GPT-2
- Required 3.14×10^{23} calculations to train
- More than all world's computers performed in 2010

- GPT3 exhibited capabilities that weren't programmed or explicitly trained for

- Emergent Abilities:
  - Learning new tasks from just examples
  - Reasoning about problems never seen before
  - Creative writing and problem-solving

Human Feedback - Teaching AI Judgment

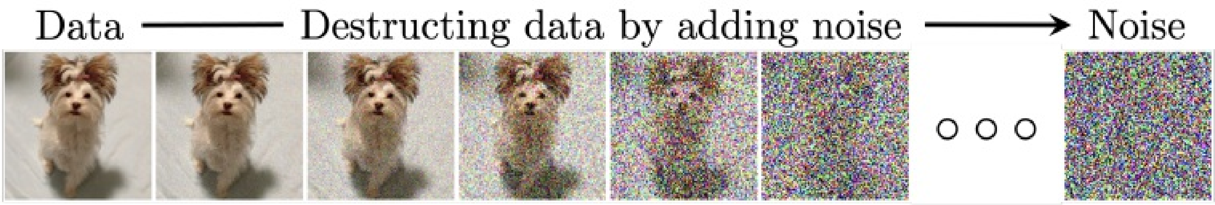
- Raw language models like GPT-3 were brilliant but dangerous
  - Completed any prompt, including harmful content
  - Conspiracy theories, illegal instructions
  - No moral judgment or safety filters

- RLHF Solution (Reinforcement Learning from Human Feedback):
  1. Humans demonstrated good responses for thousands of prompts
  2. Reward model learned to score responses like humans
     - Reward a model when it provide an answer that is aligned with human preference and penalize it otherwise.
  3. Language model fine-tuned to maximize these rewards
- Result: GPT-3 → ChatGPT transformation

Diffusion Models - Visual Creativity Revolution

Counterintuitive Concept:

- Show millions of photos with gradually added static
- Train neural network to remove noise from images



Diffusion Models: A Comprehensive Survey of Methods and Applications

Reverse Process:

- Start with pure noise
- Gradually denoise following learned patterns
- Result: Image generation from text descriptions

DALL-E 2 & Stable Diffusion (2022): "An astronaut riding a horse in the style of Van Gogh" → Photorealistic result

Questions for Discussion

- Technical Questions:

- Which phase do you think was most crucial for today's AI?
- What are the implications of emergent capabilities?
- How should organizations prepare for AI transformation?
- How do we ensure AI development benefits everyone?
- What skills become more valuable in an AI-augmented world?
- How do we navigate the "Can AI do this?" vs "Should AI do this?" question?