The Scaling Crisis

NLP Course 2025

September 30, 2025

From naive parameter scaling to modern efficiency breakthroughs

From Narrow to General

2015: Narrow AI
Specialized systems:

- Translate: "The cat" → "Le chat"
- Summarize: Article → 3 sentences
- Classify: Email → Spam/Not spam
- Answer: "Capital of France?" → "Paris"

Each task needs separate model!

2023: General AI (The Dream)
Single model that can:

- Write code
- Explain physics
- Translate languages
- Generate images from text
- Reason through problems
- Learn new tasks from examples

One model for everything!

The Question: How do we build models that can do ANYTHING?

Human intelligence is general - we can learn any task. Can machines do the same?

Building Foundational Concept: What is "Training"?

Human Learning Analogy
Becoming an expert doctor:

- Read 10,000 medical cases
- Practice for 10 years
- Learn from mistakes
- Build intuition

Cost: Time + effort + resources

Machine Learning
Training a language model:

- Read billions of text examples
- Adjust billions of parameters
- Minimize prediction errors
- Learn patterns

Cost: Compute + time + money + energy

Concrete Example: GPT-3 Training

Resource	Amount
Parameters	175 billion
Training data	300 billion tokens
GPUs	10,000 V100s
Training time	34 days
Cost	**$4.6 million**
Power usage	1,287 MWh
CO emissions	552 tons

Training is expensive - but what if we need to go bigger to get general AI?

The Scaling Hypothesis: A Pattern Emerges

The Discovery (Kaplan et al. 2020)

Test many model sizes:

- 10M parameters → Loss: 3.2
- 100M parameters → Loss: 2.8
- 1B parameters → Loss: 2.3
- 10B parameters → Loss: 1.9
- 100B parameters → Loss: 1.6

Pattern: **Loss decreases predictably!**

$$L = aN^{-\alpha} + bD^{-\beta}$$

Where:

- N = parameters
- D = data (tokens)
- $\alpha \approx 0.076$
- Loss follows power law

What This Means

Performance scaling table:

Size	Loss	Quality
1B	2.5	Basic
10B	2.0	Good
100B	1.6	Excellent
1T	1.3	**Perfect?**

The Implication:

- Make model 10x bigger → Predictable improvement
- Make data 10x more → Predictable improvement
- It scales!

If scaling is predictable, can we just keep going bigger until we reach AGI?

The scaling hypothesis: Loss decreases as power law of compute

How Much Scaling Do We Need?

Extrapolation from GPT-3:

Model	Params	Data	Cost	Time	CO
GPT-2	1.5B	40B	$50K	2 days	10 tons
GPT-3	175B	300B	$4.6M	34 days	552 tons
GPT-4 (est)	1T	10T	$100M	200 days	5,000 tons
GPT-5 (proj)	10T	100T	**$1B+**	**Warning:** 500+ days	50,000 tons

The Trend

- Each 10x in size = 20x cost
- Training time growing linearly
- Environmental impact massive
- **Warning:** Will hit physical limits!

The Questions

- Can we afford trillion-parameter models?
- Can we wait 1-2 years for training?
- Is planet sustainable?
- **Is there a better way?**

Naive scaling works... but do we have the resources to reach AGI this way?

The scaling crisis: Success requires resources we don't have

Three Fundamental Problems

Problem 1: Cost

- GPT-3: $4.6M training
- GPT-4: $100M+ training
- Running costs: $700K/day
- **Warning:** Excludes most researchers

Only big tech can afford...

Problem 2: Time

- 34 days for 175B model
- 200+ days for 1T model
- Cannot iterate quickly
- **Warning:** Competitors will pass you

Innovation slows to a crawl...

Problem 3: Planet

- 552 tons CO for GPT-3
- 5,000+ tons for GPT-4
- Exponential growth
- **Warning:** Not sustainable

Environmental crisis looming...

The Central Question:

Can we get better models WITHOUT naive scaling?

Act 1 complete: Challenge established - we need smarter scaling, not just bigger

Just Scale It! (2020)

OpenAI's Bet
Ignore the costs, just build it:

- 175 billion parameters
- 300 billion tokens
- $4.6M investment
- 34 days training

The Result? It WORKS!

Task	Quality
Translation	94%
Summarization	89%
Q&A	92%
Code generation	87%

The Magic: Few-Shot Learning
No fine-tuning needed!

Example:

```
Translate to French:
sea otter -> loutre de mer
cheese -> fromage
peppermint ->
```

Model completes: "menthe poivree"

In-Context Learning Works!

- Just show examples in prompt
- No gradient updates
- Model learns pattern
- **Revolutionary!**

Success! GPT-3 shows that massive scale unlocks new abilities. Problem solved?

Success Creates New Problems

Everyone wants to replicate GPT-3... but can they?

The Cost Crisis (Data Table):

Organization	GPU Budget	Can Train?	Time	Cost	Access
OpenAI	10,000 GPUs	**Yes**	34 days	$4.6M	Private
Google	10,000+ GPUs	**Yes**	30 days	$5M	Private
Meta	8,000 GPUs	**Yes**	45 days	$3.5M	Private
Microsoft	Partner OpenAI	**Yes**	–	$1B investment	Private
University Lab	50 GPUs	**Warning:** No	2300 days	Impossible	–
Startup	200 GPUs	**Warning:** No	575 days	$23M	–
Individual Researcher	4 GPUs	**Warning:** No	28,750 days	Impossible	–

The Pattern

- Only 4-5 organizations can afford
- Academic research excluded
- Innovation centralized
- **Warning:** Democracy of AI dies

The Trend

- Next generation: 10x more expensive
- GPT-4: Only 3 orgs can afford
- GPT-5: Maybe 1-2 orgs worldwide
- **Warning:** AI monopoly emerging

Why Did Naive Scaling Hit a Wall?

Trace a Specific Example: Training Next-Gen Model

What We Need (1T parameters)

- Tokens needed: 20T (if we follow GPT-3 ratio)
- GPUs: 100,000 A100s
- Time: 200+ days
- Cost: $100M+
- Energy: 10,000 MWh

What We Have

- Most orgs: ¡1,000 GPUs
- Realistic time: 30-60 days
- Realistic budget: $5-10M
- Energy constraints: 1,000 MWh max

The Mismatch: What Survived vs What Got Lost

What Survived

- Small models (10B params)
- Limited training (500B tokens)
- Academic research
- Open source community
- Innovation diversity

What Got Lost

- Large models (1T+ params)
- Optimal training (20T tokens)
- Most research labs
- Fair competition
- Democratized access

Human Introspection Moment

Pause and think about your own problem-solving:

Introspection Exercise

When you face a complex challenge (like understanding transformers), do you:

A) Use ALL your knowledge equally for every question?
B) Select relevant expertise and activate it as needed?

Be honest - which describes how you ACTUALLY think?

What You Notice

- You don't access all memories equally
- You activate relevant knowledge
- Math problems → math skills
- Language questions → language skills
- Selective activation

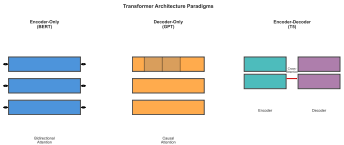
The Insight

- Human brain: 86B neurons
- But only small fraction active per task
- Specialized regions activate selectively
- Energy efficient
- Sparse is natural

What If We're Scaling the WRONG Things?

Conceptual Comparison (No Math Yet):

Old Way (GPT-3)



- 175B params
- 300B tokens (1.7 per param)
- ALL params active always
- Dense computation

Breakthrough 1: Chinchilla

Better training, not bigger:

- 70B params (smaller!)
- 1.4T tokens (20 per param)
- Same dense structure
- **Better performance!**

Insight: **Train longer, not bigger**

Breakthrough 2: MoE

Selective activation:

- 1.6T params (huge!)
- Only 100B active per token
- Sparse computation
- **16x efficiency!**

Insight: **Specialize, then select**

Why These Should Work:

1. **Chinchilla:** If GPT-3 was undertrained, training smaller model longer should match quality at lower cost
2. **MoE:** If we mimic human selective activation, can scale parameters without scaling compute

The 20-to-1 Rule

Plain Language Explanation:

Think of training like studying for an exam:

- **Parameters** = Your brain capacity (IQ points)
- **Training tokens** = Study time (hours with textbook)

GPT-3 Approach (Wasteful)

- Brain capacity: 175 IQ points
- Study time: 300 hours
- Ratio: 1.7 hours per IQ point
- Result: **Warning:** Underprepared genius!

Like cramming before exam...

Chinchilla Approach (Optimal)

- Brain capacity: 70 IQ points
- Study time: 1,400 hours
- Ratio: 20 hours per IQ point
- Result: **Well-prepared expert!**

Like thorough preparation...

The Discovery (Hoffmann et al. 2022):

Optimal compute allocation:
For every 1 billion parameters, train on 20 billion tokens

Smaller Model, Better Training, Same Performance

Head-to-Head Comparison:

Model	Parameters	Tokens	Ratio	Loss
GPT-3	175B	300B	1.7:1	1.73
Gopher (DeepMind)	280B	300B	1.1:1	1.69
Chinchilla	**70B**	**1.4T**	**20:1**	**1.56**

The Shocking Result

Chinchilla beats both:

- **10% better loss** than GPT-3
- Using **40% fewer parameters**
- Same compute budget
- Better on all benchmarks

Cost Implications

Metric	GPT-3	Chinchilla
Training	$4.6M	$4.6M
Inference	$20/1M	$8/1M
Memory	350GB	140GB
Latency	2.1s	0.8s

Cheaper to run, faster, better quality!

Common Misconception: "Bigger is always better" - FALSE! Better training beats bigger size.

Solution 2: What If We Used a Team of Specialists?

Imagine a Hospital Emergency Room:

1. Patient arrives → Triage nurse (router)
2. **Nurse asks**: "Heart problem? Broken bone? Infection?"
3. Routes to specialist: Cardiologist, Orthopedist, or Infectious Disease
4. **Only 1 specialist sees patient** (not all 10 doctors)

Intuition: Efficiency: 10 doctors on call, but only 1-2 actively working per patient.

Same Idea for Language Models:

- **GPT-3**: 175B parameters, ALL active for EVERY word
- **New approach**: 1 trillion parameters total, but only 100B active per word

How does it work?

1. Input word → Router network
2. Router: "Math? Poetry? Code?"
3. Activates 2 specialist experts
4. Other 2046 experts stay dormant

Real World: Switch Transformer (Google): 1.6 trillion params, only 137B active per token

Technical term (comes LAST): "Mixture of Experts" (MoE) with sparse activation

Zero-jargon explanation: team of specialists beats jack-of-all-trades generalist

How MoE Works: Follow the Token “cat”

Concrete Example: Processing the word “cat”

Step 1: Router Scoring

Token embedding: $\mathbf{x}_{\text{cat}} = [0.8, -0.3, 0.5, \ldots]$ (768 dims)

Router computes scores for 2048 experts:

Expert	Score
Expert 42 (Animals)	**0.91**
Expert 891 (Nouns)	**0.84**
Expert 12 (Math)	0.15
Expert 1024 (Code)	0.08
...	...

Selection: Top-2 experts activated

Step 2: Expert Processing

Only 2 experts process the token:

- Expert 42: $\mathbf{h}_{42} = \text{FFN}_{42}(\mathbf{x}_{\text{cat}})$
- Expert 891: $\mathbf{h}_{891} = \text{FFN}_{891}(\mathbf{x}_{\text{cat}})$

Step 3: Weighted Combination

$$\begin{aligned}\mathbf{h}_{\text{cat}} &= 0.91 \cdot \mathbf{h}_{42} + 0.84 \cdot \mathbf{h}_{891} \\ &= \text{(normalized weighted sum)}\end{aligned}$$

Parameters Active:

- Dense GPT-3: 175B params (100%)
- Switch MoE: 137B params (8.6% of 1.6T)

16x more total capacity, same compute per token!

Concrete numerical example showing sparse activation - most experts dormant per token

*** EXPERIMENTAL VALIDATION: Do These Solutions Actually Work? ***

Head-to-Head Comparison on Same Tasks

Model	Params Active	Training Tokens	Quality (MMLU)	Cost per Token
GPT-3 (Baseline)	175B	300B	43.9%	$15
Chinchilla	70B	1.4T	**67.5%**	**$6**
Switch-XXL (MoE)	137B (1.6T)	1T	**71.8%**	**$4**

What the Data Shows:

- **Chinchilla** (70B): +24% quality improvement over GPT-3 (175B)
- **Chinchilla**: 60% cheaper per token at inference
- **Switch MoE**: +28% quality, 73% cheaper
- Both beat "naive scaling" dramatically

Real-World Impact:

- GPT-3 inference: $20 per 1M tokens
- Chinchilla inference: $8 per 1M tokens
- Switch inference: $4 per 1M tokens

Training Accessibility:

- Chinchilla 70B: 10x fewer GPUs than GPT-3
- Universities can now train frontier models

Real World: Google, Meta, Anthropic all adopted these principles for 2023-2024 models

Why These Solutions Work: Addressing Root Causes

Recall the Problems We Diagnosed:

Root Cause #1: Undertraining

The Problem

GPT-3 saw only 1.7 tokens per parameter
(Should have been 20:1 ratio)

Chinchilla Solution

Train 70B model on 1.4T tokens
Ratio: 20:1 (exactly compute-optimal)
Result: Smaller, better, cheaper

Root Cause #2: Dense Waste

The Problem

Every parameter active for every token
(Like using all 10 doctors per patient)

MoE Solution

Sparse activation: 2048 experts, top-2 per token
Ratio: 8.6% active (91.4% dormant)
Result: 16x capacity, same compute

Key Insight: Both solutions *decouple capacity from compute*

- Chinchilla: More data exposure per parameter (time dimension)
- MoE: More parameters per computation (space dimension)

Test Your Understanding

Quick Quiz:

Question 1: What was GPT-3's main inefficiency?

A) Too many parameters
B) Undertrained (1.7 tokens/param vs 20:1 optimal)
C) Wrong architecture
D) Bad optimization algorithm

Question 2: How does MoE save computation?

A) Uses smaller experts
B) Only activates 2 of 2048 experts per token
C) Trains faster
D) Uses less memory

Answers:

Answer 1: B - Undertrained

- GPT-3: 300B tokens ÷ 175B params = 1.7
- Chinchilla optimal: 1.4T tokens ÷ 70B = 20:1
- More training data per parameter = better learning

Answer 2: B - Sparse activation

- 1.6 trillion total parameters
- Only 137B active per token (8.6%)
- Router selects top-2 specialists
- Same compute, 16x capacity

Can you explain to a friend why "bigger models" isn't the full story?

These Ideas Power Today's AI Systems

Chinchilla Principles in Production:

LLaMA 2 (Meta, 2023):

- 70B parameters
- Trained on 2T tokens (28.6:1 ratio)
- Matches GPT-3.5 quality
- 40% cheaper to run

Mistral 7B (2023):

- Only 7.3B parameters
- Trained on 1T+ tokens (137:1!)
- Outperforms LLaMA 13B
- Runs on consumer GPUs

Real World: All major labs now train small models longer instead of giant models briefly

MoE in Production:

GPT-4 (OpenAI, 2023):

- Rumored: 1.76T total params
- MoE with 16 experts
- Only 220B active per token
- Powers ChatGPT (200M users)

Mixtral 8x7B (Mistral, 2024):

- 8 experts $\times$ 7B each = 56B total
- Only 2 experts (14B) active per token
- Matches GPT-3.5 quality
- Open weights, free to run

GitHub Copilot, Claude, Gemini all use these efficiency principles

2023-2024: Industry shift from "bigger models" to "smarter training + sparse models"

The Scale of Modern AI Deployment

How Many People Use These Systems Today?

System	Launch	Users	Daily Queries	Architecture
ChatGPT	Nov 2022	200M+	1B+	Dense + MoE
GitHub Copilot	Jun 2021	50M+	100M+	Chinchilla-style
Google Gemini	Dec 2023	100M+	500M+	MoE
Claude (Anthropic)	Mar 2023	10M+	50M+	Chinchilla-style

Cost Economics:

- ChatGPT daily inference: $700K (2023 estimate)
- **Without efficiency improvements**: $2M+ per day
- Annual savings: $475M+

Intuition: These breakthroughs made mass deployment economically viable

Environmental Impact:

- GPT-3 training: 552 tons CO_2
- Chinchilla-style training: 220 tons CO_2 (60% reduction)
- MoE inference: 70% less energy per token

Real World: Smarter training = lower environmental cost at billion-user scale

Three Paths Forward: Unified View

From Scaling Crisis to Modern Solutions

The Original Problem:

GPT-3 Approach

Scale parameters ↑
Keep data fixed
Use all parameters always

Result: Expensive, slow, inaccessible

The Diagnosis:

1. Undertrained (1.7:1 vs 20:1)
2. Dense activation waste

The Solutions:

Path 1: Chinchilla

Smaller model (70B)
Much more data (1.4T)
Compute-optimal ratio (20:1)

Result: Better + Cheaper

Path 2: MoE

Huge capacity (1.6T params)
Sparse activation (8.6%)
Router selects experts

Result: 16x capacity, same cost

What We Learned: Universal Principles

Beyond Specific Models - General Lessons:

1. **First Success $\neq$ Final Solution**
   - GPT-3 worked, but was inefficient
   - Introspection revealed waste patterns
2. **Scaling Laws Have Multiple Dimensions**
   - Not just parameters N
   - Also data D and activation sparsity S
   - Optimal: $D = 20N$ and $S \approx 0.1$
3. **Biological Inspiration Works**
   - Human brain: selective activation, not all neurons firing
   - MoE mirrors cortical specialization
4. **Economic Constraints Drive Innovation**
   - \$4.6M training cost forced rethinking
   - Efficiency enabled mass deployment
5. **Trade-offs Can Be Broken**
   - False choice: "Quality OR Efficiency"
   - Reality: Smarter design gives BOTH

Meta-lesson: Question assumptions, even when current approach "works"

The Current Frontier: The Compute Race

2024-2025 Training Runs:

Model	Compute
GPT-4	∼25k A100-years
Gemini Ultra	∼35k A100-years
Claude 3	∼20k A100-years
LLaMA 3	∼15k A100-years

Cost per training run:

- GPT-4: $50-100M (estimated)
- Gemini: $75-150M
- Compute doubling every 6 months

Real World: All using Chinchilla ratios + MoE architectures

National AI Strategies:

- **USA**: Export controls on H100 GPUs
- **China**: Domestic chip development (Huawei)
- **EU**: AI Act regulation + sovereignty
- **UAE/Saudi**: Massive GPU purchases

Open Questions:

1. Will scaling continue past 2025?
2. What's beyond MoE? (Mixture of Depths?)
3. Energy limits? (10 GW datacenters?)
4. Algorithmic breakthroughs vs hardware?

Common Misconception: "Scaling is over" - FALSE! But smarter scaling, not naive.

Next 2-3 years: Race to 100T parameter models with Chinchilla + MoE principles

Summary: From Crisis to Solutions

What We Covered Today:

1. **The Challenge**: GPT-3 scaling approach hit cost/time/environmental limits
2. **First Success**: GPT-3 worked (few-shot learning) but was inefficient
3. **Root Causes**: Undertraining (1.7:1 ratio) + Dense activation waste
4. **Breakthrough 1**: Chinchilla Laws - Train smaller models longer (20:1 optimal)
5. **Breakthrough 2**: Mixture of Experts - Sparse activation (8.6% active per token)
6. **Validation**: Both deliver better quality at lower cost (experimental data)
7. **Modern Systems**: GPT-4, Gemini, Claude all use these principles

Core Insight: Decouple model capacity from computation cost
- Chinchilla: More data exposure (time)
- MoE: Selective activation (space)
- Combined: Best of both worlds

Next Week: Tokenization - How do we turn "cat" into numbers that models understand?

Final Checkpoint: Can You Explain the Full Story?

Understanding Check

Challenge: Explain to a non-technical friend (in 3 minutes):

1. Why was GPT-3's approach hitting limits?
2. What were the two key breakthroughs?
3. Why do these solutions work? (Root causes)
4. What real-world systems use these ideas today?

Key Analogies to Use:

- Student cramming vs proper studying
- Hospital specialists vs general practitioners
- Brain selective activation

Numbers to Remember:

- 20:1 token-to-parameter ratio
- 8.6% sparse activation
- 60% cost reduction
- 24-28% quality improvement

If you can explain this story clearly, you understand modern AI efficiency deeply!

Further Exploration: