

Foundational ideas of modern AI

Sutskever's List

Rich Heimann



MEAP



MANNING



MEAP Edition
Manning Early Access Program

Sutskever's List
Foundational ideas of modern AI
Version 1

Copyright 2025 Manning Publications

For more information on this and other Manning titles go to manning.com.

welcome

Thank you for joining the MEAP for *Sutskever's List*. This book is written for readers with a working familiarity with AI concepts and a curiosity that extends beyond the algorithms into the history, strategy, and culture that have shaped modern machine learning. You do not need to be an academic researcher, but some prior exposure to deep learning architectures, whether from hands-on work, reading research summaries, or following AI news, will help you get the most from it.

The book takes you inside a curated set of research papers known informally as "*Sutskever's List*," which Ilya Sutskever once described to John Carmack as covering "90% of what matters today" in AI. However, instead of viewing each paper as a separate contribution, the chapters depict them as interconnected threads within the broader narrative they represent. You will see how technical breakthroughs intersect with organizational decisions, safety debates, and cultural inflection points, and how they continue to shape the systems we use today.

Moreover, this is not just a book about what these models and methods do but as clues to Sutskever's worldview. It is about why they emerged when they did, what problems they solved, how they interacted with other advances, and what they revealed about the most influential AI researcher of our time. My aim is to give you both a conceptual toolkit for practice and a richer sense of the forces, technical and human, that have driven AI's evolution.

Your feedback will help make this the strongest book it can be. Please share your questions, comments, and suggestions in the [liveBook Discussion forum](#). I look forward to hearing what resonates, what needs more clarity, and what you think should be expanded.

—Rich Heimann

brief contents

- 1 What Did Ilya See?*
- 2 The AlexNet Moment*
- 3 Residual Revolution (ResNet)*
- 4 Deep Learning Accelerates*
- 5 Attention is All You Need (Transformers)*
- 6 The Birth of Hyperscale (Scale)*
- 7 Pretraining is dead: The Pivot to Reasoning*
- 8 Simplicity in Complexity (MDL/Kolmogorov)*
- 9 Safe Superintelligence (AGI/Safety/Alignment)*

1 What Did Ilya See?

This chapter covers

- Ilya Sutskever's journey to prominence
- The origin and significance of Sutskever's List
- The staged release of GPT-2
- The importance of the cultural impact of GPT-2
- Ethical and safety concerns raised by GPT-2

In November 2023, Elon Musk tweeted, “*Something scared Ilya enough to want to fire Sam.*[\[1\]](#) *What was it?*” Soon after, the cryptic phrase “*What did Ilya see?*” went viral, capturing the imagination of AI researchers and enthusiasts alike.[\[2\]](#)[\[3\]](#) Had Sutskever glimpsed an existential risk that the rest of the world had missed? Many believed he had. They believed that removing OpenAI CEO Sam Altman was necessary for the good of the company, and perhaps even humanity. In that moment, OpenAI’s reclusive Chief Scientist went from respected researcher to enigmatic guardian of knowledge. Yet, this wasn’t the first time Sutskever’s guardedness sparked controversy. In 2019, OpenAI famously declared its new language model, GPT-2, “too dangerous to release.” This move ignited debate about artificial intelligence safety, ethics, and transparency.

The question “*What did Ilya see?*” and GPT-2’s controversial designation reflect to many a figure perceiving something unsettling at AI’s frontier. Yet, understanding these fascinating episodes requires us to appreciate the broader intellectual landscape and the context from which these moments emerged. To do so, we must explore the foundational ideas that shaped Ilya’s thinking and the field. We must analyze the eponymous “Sutskever’s List.” The list, which Sutskever once described to the legendary software developer John Carmack as containing “90% of what matters today.”[\[4\]](#)

Such clarity, however, is elusive.

Instead of viewing each paper as a separate contribution, the chapters depict them as interconnected threads within the broader narrative they represent. In doing so, this book targets readers who seek more than just independent summaries; it is for those who, like me, wish to understand how these ideas connect, their significance at the time, their enduring relevance today and what they collectively reveal about the most transformative period in the history of artificial intelligence. In this way, the book is more than an anthology.

The book analyzes the technical details of each paper while weaving together technical milestones, cultural moments, paradigm shifts, blind spots, dogma, and the field's ambitions and anxieties. This synthesis transforms what might otherwise be a daunting reference with fragmented insights into a clear and cohesive narrative. Thus, the book is narrative first, encyclopedic second.

As the book title suggests, we will follow Ilya's intellectual evolution and the shifting perspectives that define his thinking. Each paper serves as a clue: a window into what Ilya believes matters most. After all, these aren't just important papers; they create a conceptual scaffolding for one of the field's most influential thinkers, with over 600,000 citations. In tracing this list, we'll explore the core themes of Ilya's thinking, including the tensions between accelerationism and caution, scaling and alignment, success and failure, and ambition and uncertainty. *Sutskever's List* reveals as much about its compiler as it does about its code.

Although the book focuses on Ilya, its purpose is not biographical. Instead, the book uses Ilya's intellectual journey as a narrative throughline, illustrating how his decisions—from AlexNet and Google Brain to the founding of OpenAI, the pivot to safety, and his eventual departure—highlight key inflection points in the entire field. Crucially, Ilya's experiences and intellectual journey not only intersect with modern AI; they also reflect it.

Chapter One traces Ilya's rise from a respected researcher to one of AI's most enigmatic and influential thinkers. It presents the mysterious "Sutskever's List" and outlines Ilya's mental model. The chapter concludes by discussing GPT-2, a model notably absent from the List, precisely because it vividly illustrates the real-world stakes of Ilya's work and philosophy. When GPT-2 was declared "too dangerous to release," the tension between nascent scaling and safety became apparent for the first time. This tension would deepen, shaping OpenAI's culture and eventually erupting into the public crisis. We discuss GPT-2 not because it's where the story begins or ends, but because it's where it changes.

Only after we've glimpsed some of the tensions at the heart of modern AI can we fully appreciate the breakthroughs discussed later, such as AlexNet, Deep Speech 2, ResNet, Neural Turing Machines, and Transformers, that appear on *Sutskever's List*. Chapter One does not dissect these papers. Instead, it provides narrative context and explores the cultural landscape from which these innovations emerged, clarifying precisely what is at stake by starting somewhere in the middle. As we move forward in the book, the intensity and urgency of this opening chapter will inevitably give way to more technical discussions. Yet it is this context that lends necessary weight and meaning to those detailed explorations. Moreover, exploring GPT-2 also brings us back to our opening question: What did Ilya see? As we progress through the List itself, we may uncover what he saw, why he viewed it with such caution, and why his prescient or perhaps overly cautious vision resonates today.

1.1 Ilya's Rise

Ilya Sutskever pursued graduate studies at the University of Toronto, studying under the so-called “godfather of AI,” Geoffrey Hinton.[\[5\]](#) At the time, Hinton was widely respected but had not yet reached the legendary status he holds today, a status secured in part by Sutskever’s work. In September 2012, Ilya, alongside Alex Krizhevsky and Hinton, trained an artificial neural network to classify images. Their landmark paper stunned the AI world by dramatically reducing error rates in the emerging ImageNet Large Scale Visual Recognition Challenge (ILSVRC).[\[6\]](#) The team decimated the competition.

With this single achievement, decades of skepticism largely evaporated, and Sutskever became a key figure at the heart of the deep learning revolution. This victory was the empirical evidence skeptics could not ignore.[\[7\]](#) When the results were discussed, even Yann LeCun, a longtime advocate of CNNs, was astonished. He called the victory “an unequivocal turning point in the history of computer vision.”[\[8\]](#) Decades of hand-crafted vision approaches had been toppled by an artificial neural network trained end-to-end on raw images.[\[9\]](#) AlexNet, the architecture behind this breakthrough, is the subject of the next chapter, where we examine its innovations and impact in greater depth.

In 2013, Google recognized the significance of the team’s work and acquired DNNresearch, the startup founded by Hinton and his students, including Sutskever and Krizhevsky.[\[10\]](#) The acquisition allowed Google to internalize the expertise just as deep learning was beginning to reshape the field.[\[11\]](#) Ironically, this acquisition would help motivate the founding of OpenAI, an organization explicitly created to counterbalance Google’s growing influence.

At Google Brain, Sutskever became central to advancing deep learning, contributing significantly to Google’s large-scale AI infrastructure.[\[12\]](#)[\[13\]](#) He was an early collaborator on TensorFlow, Google’s open-source deep learning library, and co-authored the seminal AlphaGo paper.[\[14\]](#) Moreover, several projects on *Sutskever’s List* emerged directly from his years at Google, including co-developing the sequence-to-sequence learning framework, pioneering recurrent neural network regularization, and indirectly supporting influential architectures such as Pointer Networks and Neural Turing Machines.[\[15\]](#)

Yet, by 2015, a new opportunity was taking shape. Sam Altman, then president of Y Combinator, invited Sutskever to a private dinner with Elon Musk and others.[\[16\]](#)[\[17\]](#) Unbeknownst to Sutskever, he was the guest of honor. The dinner conversation centered on the future of AI, including whether incumbents like Google were unbeatable or if a new research lab could be created as a “counterbalance.”

The idea of a “counterbalance” had gained urgency in the weeks prior. At a party, Musk had clashed with Larry Page, Google’s co-founder. Page shared his vision for a future where humans and machines compete for resources. Musk responded bluntly, “If that happens, we’re doomed.” Page dismissed the concerns and called Musk a “specieist.”[\[18\]](#) Shaken by Page’s casual dismissal of human extinction and his surreal claim that preferring humans to machines amounted to speciesism, Musk wanted to build a counterweight to Page’s vision.

The whole idea resonated with Sutskever. On *The Robot Brains* podcast, Ilya recalled, “I really wanted it to be engineering-heavy. And seeing that Elon was going to be involved, I thought, well, who better? I couldn’t imagine a better person from whom to learn the big engineering project side of things.”^[19] After some hesitation and months of counteroffers, Sutskever left Google. Musk later described Ilya as the “linchpin” of the new venture, indicating how critical his involvement was to the project’s credibility.^{[20],[21]}

OpenAI was founded in late 2015. Ilya became OpenAI’s Chief Scientist and the “soul of the company and its driving force in research,” as one account later described, shaping OpenAI’s research agenda and culture.^{[22],[23],[24]} Sutskever’s influence ensured that the work remained ambitious, prompting bold, even grandiose claims, such as building *“systems that can do things humans were not capable of doing before.”*^[25] While bold, this statement was vague and lacked an articulated technical goal. In fact, much of this early period was marked by philosophical debates rather than concrete plans. OpenAI discussed balancing openness with safety and debated the pace at which it should pursue artificial general intelligence (AGI). These debates reveal an organization shaped by broad aspirations rather than precise strategic direction.

Yet, the discussions weren’t theoretical for long. OpenAI soon began translating its philosophical aspirations into practical milestones. The lab developed OpenAI Gym, a platform to accelerate reinforcement learning research.^[26] They also achieved early breakthroughs in AI-driven game-playing, notably mastering the complex multiplayer game Dota 2.^[27] By 2018, these efforts converged into focused experiments with Transformers, an architecture ironically introduced by Google, culminating in the release of a language model called GPT-1. This shift from lofty uncertainty to targeted exploration marked a pivotal moment for OpenAI. It also coincided with an internal reconfiguration, as Elon Musk, a co-founder and key backer, parted ways with OpenAI due to conflicts with his role at Tesla.^[28]

Yet, behind the scenes, it emerged that Musk proposed taking more direct control of OpenAI due to concerns that it was falling behind Google.^[29] However, Altman and the other co-founders rejected Musk’s bid.^[30] Consequently, Musk withdrew from the venture, leaving the remaining leadership, Altman and Sutskever, firmly in charge of OpenAI’s direction. Sutskever chose to stand by the nonprofit vision and Altman’s leadership rather than cede control to Musk, reflecting a principled stance that the company should remain focused on its long-term safe AGI goals.

OpenAI’s research ramped up. Sutskever oversaw the development of GPT-2 but decided *not* to release the model’s full version.^[31] By 2020, OpenAI fully embraced *scaling* and began training larger models.^[32] In June 2020, OpenAI introduced GPT-3, a 175-billion-parameter language model that stunned many observers with its versatility in generating text. GPT-3 required massive engineering. Sutskever guided the vision for such large models and helped develop the infrastructure and team to train them. He is widely credited as a key architect of these advances; by one account, Sutskever “played a key role in the development of ChatGPT” and the GPT series models behind it.^[33]

As OpenAI’s creations grew more interesting, Sutskever’s reputation as one of the foremost research minds in the field grew. This period solidified OpenAI’s reputation as a leader in AI and Sutskever’s as a visionary and a bit of a recluse: “I lead a very simple life,” Ilya said in one interview. “I go to work; then I go home. I don’t do much else. There are a lot of social activities one could engage in, lots of events one could go to. Which I don’t.”^[34]

Yet, during this time, Sutskever also became more publicly reflective about the implications of OpenAI's research. In February 2022, nearly a year before ChatGPT was released, Ilya generated significant debate by posting a curious statement on Twitter (now X): "It may be that today's large neural networks are slightly conscious."[\[35\]](#) The tweet was provocative and broached a topic many researchers treat skeptically: machine consciousness. Sutskever offered no further explanation and did not delete the remark, even after it was met with criticism and incredulity.

While some researchers, like Andrej Karpathy, supported the assertion by responding with "agree...consciousness is a useful insight for compression," others were less supportive. Turing Award winner Yann LeCun responded with "nope," and Murray Shanahan, a Professor at Imperial College London and Principal Scientist at Google DeepMind, added "in the same sense that it may be that a large field of wheat is slightly pasta."[\[36\]](#),[\[37\]](#),[\[38\]](#) The episode highlighted Sutskever's willingness to voice unconventional thoughts. While most experts dismissed the "slightly conscious" idea, the comment foreshadowed Ilya's increasing focus on AI's philosophical and safety dimensions.

Increasingly, Ilya became a spiritual evangelist for safe artificial general intelligence (AGI) within the company. OpenAI launched a new "Superalignment" team co-led by Ilya Sutskever and researcher Jan Leike. The team's goal was to steer a potentially superintelligent AI and prevent it from "going rogue" within four years. In a widely read blog post published eight months after the initial release of ChatGPT, Sutskever stressed urgency, suggesting that while superintelligence might feel distant, it "could happen this decade."[\[39\]](#),[\[40\]](#)

Reports from this period portray him as a passionate, if eccentric, figure rallying the company around the mission of creating beneficial AGI. He reportedly encouraged employees to "feel the AGI," thus treating the quest for artificial general intelligence with an almost reverential fervor.[\[41\]](#) In one notable anecdote, Sutskever commissioned a wooden effigy representing an "unaligned" superintelligent AI that could harm humanity and ceremonially burned it before his colleagues.

These ritualistic displays, described in an exposé of OpenAI's internal culture, were unusual even by tech industry standards. They illustrate Sutskever's intense conviction and the sense of gravity he brought to the alignment task. Some insiders saw this as inspirational, while others found it worrying.[\[42\]](#) Either way, by late 2023, Sutskever had clearly shifted toward a more urgent, cautionary stance that would soon compel him to confront OpenAI's trajectory head-on.

Tensions came to a head on November 17, 2023, when the OpenAI Board, of which Sutskever was a member, abruptly ousted Sam Altman.[\[43\]](#) Sutskever was reportedly a key instigator behind this decision.[\[44\]](#) Publicly, the Board attributed Altman's dismissal to a vague "breakdown of communications," an ironically uncommunicative phrase that barely concealed a deeper strategic rift.[\[45\]](#) In Ilya's view, OpenAI was moving too fast and might soon cross a dangerous threshold if left unchecked.

Immediately following the firing, reports emerged that just before Altman's termination, several OpenAI researchers, likely from Sutskever's alignment team, had written a letter to the Board warning of a new AI breakthrough that they believed could pose an existential threat.[\[46\]](#) They referenced a secret research project codenamed "Q*" (pronounced "Q-star"), which, according to insiders, showed an ability to solve specific math problems at a grade-school level. This hardly seems to justify the extreme actions, but some researchers felt it might scale into a more general reasoning machine and urged the Board to consider the safety implications.[\[47\]](#)

While OpenAI never confirmed this publicly, Reuters reported the existence of the Q* project and the alarm it raised as a key factor in the Board's drastic action.[\[48\]](#) The Altman firing, in retrospect, echoed the unresolved tensions first glimpsed during the GPT-2 "too dangerous to release" episode; a pivotal event that we'll explore in greater depth shortly. In both instances, OpenAI took bold, preemptive actions based on anticipated risks, not immediate threats. First, a language model capable of crafting tales about four-horned unicorns was dramatically declared "too dangerous to release." Four years later, a CEO was removed over opaque safety concerns curiously tied to an AI's newfound ability to solve grade-school math problems; a capability whose existential implications were nonobvious. The thread that connects them is a deeply held belief, championed by Sutskever, that acting too late is worse than acting prematurely. However, whereas the GPT-2 decision was largely symbolic, the Altman firing had real consequences. It transformed hypothetical concerns into an institutional crisis.

Following the event, "*What did Ilya see?*" went viral, capturing the industry's sudden existential dread.[\[49\]](#),[\[50\]](#) But as with GPT-2, the feared catastrophe never arrived, leaving OpenAI to reckon with a crisis entirely of its own making. The irony is that the impulse to guard against catastrophe caused one.

The aftermath was chaotic. OpenAI's staff were shocked. At a tense all-hands meeting, Ilya attempted to justify the decision, telling employees that the Board was "doing its duty."[\[51\]](#) However, the move backfired with no clear communication about what specific safety concern or misstep by Altman had prompted this. Within 48 hours, over 700 of OpenAI's 770 employees had signed an open letter threatening to quit and follow Altman to whichever new venture he pursued unless the Board reversed course.[\[52\]](#)

Sutskever became the target of intense criticism from what many saw as a coup that jeopardized the company's stability. Under this pressure and facing the likely collapse of OpenAI, Sutskever reversed his stance. On November 20, just three days after the Board's decision, Ilya apologized, tweeting: "*I deeply regret my participation in the Board's actions,*" adding, "I never intended to harm OpenAI."[\[53\]](#),[\[54\]](#)

The mea culpa was extraordinary: it's rare for a board member to disavow a decision, underscoring how dramatically the situation had spun out of control. For Sutskever, the episode was personally and professionally bruising. He stepped down from the OpenAI Board in the wake of the fiasco.[\[55\]](#) AI was anything but boring during these times. For months after, reports indicated Sutskever had not been actively working at OpenAI; some said he was on leave, others said he was quietly leading his team from afar, but it was evident his internal influence had waned.[\[56\]](#) The Altman saga illustrated Sutskever's profound commitment to AI safety and willingness to risk everything on principle. However, it also illustrated the perils of such a unilateral approach with a hint of religious fanaticism.

In May 2024, Ilya and OpenAI "parted ways," an outcome many saw as inevitable after the events.[\[57\]](#) Sam Altman announced Sutskever's exit in a respectful internal message, calling him "easily one of the greatest minds of our generation" and expressing sadness at his departure.[\[58\]](#) In the same week, Jan Leike, Sutskever's co-lead on the Superalignment team, also resigned, citing an erosion of trust in OpenAI's direction.[\[59\]](#),[\[60\]](#) With both leaders gone, OpenAI dismantled the Superalignment team.[\[61\]](#) Ilya's departure transformed the unease that began with GPT-2 into something personal and palatable.

1.2 The List

Legendary game developer John Carmack faced a significant challenge. After reshaping the gaming world with iconic titles like *Wolfenstein*, *Quake*, and *DOOM*, Carmack directed his intense curiosity toward artificial general intelligence (AGI). Yet, AI was uncharted territory. During a recruiting meeting with OpenAI's founders, Carmack inquired about how he could accelerate his understanding. In response, Sutskever handed him something that quickly took on a life of its own: a curated reading list that Ilya promised would deliver "90% of what matters today" in artificial intelligence.

Carmack took the challenge to heart. "And I did. I plowed through all those things, and it all started sorting out [AI] in my head."

When Carmack revealed that he had the list, it immediately captured the imagination of the AI community. The testimonial from a legendary technologist lent weight to the List's importance, which promised to cut through the noise of thousands of publications and point straight to the most pivotal ideas. Carmack was struck that the key advances were "not extreme black-magic mathematical wizardries." He added, "a lot of them are relatively simple techniques that make perfect sense to me now that I understand them."[\[62\]](#) Such comforting tautology encouraged newcomers and made AI seem approachable.

The notion that readers could access "90% of what matters" from just a couple of dozen references in a rapidly advancing field was tantalizing. The fact that Sutskever personally curated the list added to the allure and mystique, especially given his measured online presence. Despite having over 481,000 followers on X (formerly Twitter), he posts infrequently. Sutskever's reticence transforms what might otherwise be a routine reading list into a totem.

Yet, despite Carmack frequently mentioning the list in interviews, its contents remain undisclosed, adding to the mystique. As word spread on forums and social media, the excitement grew, especially following ChatGPT's popularity. ChatGPT was released in November 2022 and reached over 100 million users within two months of its launch.[\[63\]](#) This event was a catalyst, reigniting curiosity about the List. By early 2023, interest in the List had reached a fever pitch: a Hacker News thread, "What were the papers on the list Ilya Sutskever gave John Carmack?" drew more than one hundred and thirty comments of speculation and crowdsourced guesses.[\[64\]](#) So many people wanted the List that Carmack posted on X, expressing his hope that Ilya would make it public. Carmack wrote that "a canonical list of references from a leading figure would be appreciated by many."[\[65\]](#) Yet, that never happened.

The mystery sparked a treasure hunt as various AI enthusiasts sought to reconstruct the list using hints and secondhand reports. In 2023, an OpenAI researcher compiled a likely version of the list and shared it online, reigniting the conversation. It has since been viewed nearly one million times.[\[66\]](#),[\[67\]](#) This book adopts this carefully reconstructed version.

The intrigue surrounding Sutskever's List has grown over time. It has become a common cultural touchstone. "Have you read Sutskever's List?" has become shorthand for asking if someone knows the fundamentals of modern AI research, even if that person hasn't actually read the papers. Just knowing about the List carries cachet.

Today, a certain mythos clings to *Sutskever's List*, partly because Ilya has never officially published it, despite Carmack's public nudging. Paradoxically, Sutskever's silence reinforces its allure, drawing in everyone from eager enthusiasts to intrigued book publishers, some of whom, I can personally attest, found the temptation irresistible. In a field where breakthroughs arrive relentlessly, there's something uniquely comforting, even seductive, about the notion of a stable canon, quietly handed down by one of the field's grandmasters. Yet, to many, it's more than a study guide. Many look at *Sutskever's List* and try to decode his worldview. What exactly does Ilya Sutskever believe we need to understand about AI? Is the list for safety hawks? Alignment believers? Friends of AGI? Practitioners? In a word, yes.

The List includes papers defining the deep learning revolution in computer vision, such as *AlexNet* and *ResNet*. It also highlights attention-based neural networks like Pointer Networks and Transformers, representing significant architectural advancements in natural language processing. Recurrent and long short-term memory networks are also prominent, reflecting the dominant natural language processing paradigms before attention-based architectures. The list also includes Andrej Karpathy's blog, *The Unreasonable Effectiveness of Recurrent Neural Networks*, Chris Olah's influential essay, *Understanding LSTM Networks*, and Alexander "Sasha" Rush's, *The Annotated Transformer*.

The list recognizes engineering innovations and includes papers such as GPipe, which details Google's approach to pipeline parallelism for training large models. It also discusses scaling laws, including OpenAI's 2020 study, demonstrating how model performance improves as scale increases. These artifacts underscore Sutskever's interest in understanding how far brute-force scale can push the frontier and at what cost.

The List doesn't stop at engineering feats or new architectural designs. It also includes papers on minimum description length, Kolmogorov complexity, and algorithmic randomness, all topics Ilya regularly discusses in interviews. The presence of Scott Aaronson's blog, *The First Law of Complexodynamics*, and his whimsical *Coffee Automaton* signal an openness to complexity theory. In short, Sutskever's List offers a panoramic view of the converging threads that define the era. From deep learning breakthroughs in vision and language to scaling strategies and architectural innovations to theoretical reflections on complexity and intelligence. The List is a compilation of the last fifteen years of machine learning research. It's little wonder that many now treat these works as a de facto canon for serious AI practitioners, engineers, and researchers.

This book analyzes the artifacts on the List primarily by theme, though the general progression still follows a loose chronological order as we connect conceptual threads. Moreover, each artifact is given dedicated attention including its technical and cultural significance. However, some artifacts are treated more concisely than others. Rest assured that every artifact on the List will be explored, leaving no essential insights overlooked.

Additionally, while the List guides the book's core narrative, primarily covering roughly a decade from 2012 to 2022, our exploration extends beyond this window. The book selectively introduces earlier and later research to provide critical context, illuminate blind spots, and clarify Sutskever's broader philosophical perspective. However, the primary focus remains securely tethered to these core artifacts and their immediate intellectual surroundings. This book is not intended to provide a comprehensive history of artificial intelligence research.

Manning books often adopt a “mental model” approach, offering readers a clear conceptual framework shaped by an author’s experiences and insights. However, this book is different. The mental model presented here is not my own; instead, it reflects Ilya Sutskever’s worldview, which is derived directly from the themes and artifacts in his List. By unpacking Ilya’s mental model, readers gain more than historical context; they acquire a conceptual toolkit for interpreting past developments, anticipating future shifts, and guiding practical decisions.

Understanding Sutskever’s perspective equips practitioners to navigate polarized debates around AI, while equipping practitioners to recognize which architectures have enduring value. Understanding what has worked—and why—not only informs practical engineering trade-offs but also positions readers to anticipate future paradigm shifts rather than react to them. In short, the book provides conceptual clarity and, where applicable, practical insights, all viewed through the eyes of one of AI’s most influential thinkers.

With this context in mind, let’s introduce the foundational themes of Sutskever’s worldview. Consider these initial insights not as definitive conclusions, but as starting points we’ll progressively refine while exploring deeper into the ideas shaping modern AI.

Don’t Bet Against Deep Learning: Ilya explicitly believes that “one doesn’t bet against deep learning.”^[68] This belief is reflected in the List, highlighting deep learning and its successes while omitting references to older symbolic approaches and classical planning. Remarkably, reinforcement learning (RL) is also missing, despite Sutskever leading OpenAI’s early work on reinforcement learning and its role in projects like AlphaGo and OpenAI Five. Instead, the list concentrates almost solely on supervised and unsupervised learning with deep learning.

Engineering Pragmatism: The List prioritizes large-scale engineering efforts over purely theoretical innovation. Influential projects like AlexNet and Deep Speech 2 illustrate this philosophy, combining established techniques and significant computational resources to deliver unprecedented real-world performance gains. While critics dismiss this approach as merely incremental “engineering” rather than innovative “science,” Ilya argues that tangible progress arises precisely from pragmatic experimentation, iterative improvement, and real-world deployment, rather than isolated theoretical insights.

Do More with Less at Scale: Sutskever prefers scale above all else. For example, ResNets introduced a simple architectural trick called residual connections that unlocked the training of extremely deep models.^[69] Transformers replaced recurrent architectures with parallelizable attention, enabling training on unprecedented volumes of text. Even a paper like “Order Matters,” one of the more niche papers on the List, explores how rearranging sequence data can make training more efficient.

Emergence and Compression: Sutskever’s List explores Kolmogorov complexity, minimum description length (MDL), and even an essay pondering how complexity rises then falls in closed systems, all hinting at a philosophical bent in his thinking. Why would a busy AI engineer recommend reading about algorithmic randomness or the “coffee automaton”? This is likely because Sutskever views intelligence as a compression process and ultimately finding simpler, more abstract representations of complex phenomena. In this view, intelligence emerges when a system can distill raw experience into a minimal, generalizable form.

With the conceptual framing in mind, let's turn to an event that vividly embodies the tension between scale and safety. GPT-2 is a concrete expression of Ilya's emerging cautionary stance, and a primer for the profound shifts soon to reshape both OpenAI and the broader AI community.

1.3 Too Dangerous To Release

On February 14, 2019, *The Guardian* published an article titled "New AI fake text generator may be too dangerous to release, say creators."^[70] The piece went viral and stressed OpenAI's decision to withhold the full-sized GPT-2 model, citing fears of malicious use. OpenAI described GPT-2 as "so good...and the risk of malicious use so high" that it was "breaking from its normal practice" of open release and publication. The article emphasized how GPT-2 could "write convincing news stories and works of fiction." Accompanying the article was a black-and-white photograph of George Orwell, cigarette in mouth, evoking a dystopian sensibility that echoed the themes of *1984*. The reference to Orwell and the phrase "deepfakes for text" in the second line of the article establishes a connection between the emerging capabilities of generative artificial intelligence and the potential for misinformation.

GPT-2 marks a cultural and intellectual turning point for AI. It was the first time the field paused, and safety wasn't a footnote. GPT-2 is a primer for what's to come, including GPT-3, scaling laws, and emergent behavior. It also reflects what came before, when publishing was the default and sharing models were the norm.

The official OpenAI blog, "Better Language Models and Their Implications," announced: "Due to our concerns about malicious applications of the technology, we are not releasing the trained model." Instead, OpenAI provided a much smaller 124 million parameter version and a research paper as "an experiment in responsible disclosure."^[71] The full GPT-2 model, training data, and code were all withheld, representing a significant departure from OpenAI's earlier commitment to open publication and a shift towards a more guarded and selective strategy.

Interestingly, this change in posture had been anticipated. In 2018, nearly a year before the GPT-2 release, OpenAI quietly updated its Charter, reflecting an evolving perspective on its role in a more precarious technological landscape. The revised Charter stated, "We are committed to providing public goods that help society navigate the path to AGI. Today, this includes publishing most of our AI research, but we expect that safety and security concerns will reduce our traditional publishing in the future while increasing the importance of sharing safety, policy, and standards research." Consequently, GPT-2's limited release seemed less like a sudden reaction to a technological breakthrough and more like the first public signal of an institutional pivot that was already underway.

Nevertheless, OpenAI presented its decision as grounded in specific risks. The official blog post outlined several scenarios that justified a more cautious release, alongside potential applications such as writing assistance, question-answering, unsupervised translation, and improved speech recognition. These concerns included generating misleading news articles, impersonating individuals online, producing abusive or faked content for social media, and automating spam and phishing campaigns. OpenAI argued that these scenarios meant a model like GPT-2 could be dangerous in the wrong hands, especially given broader trends in AI-generated fake content. “Technologies are reducing the cost of generating fake content and waging disinformation campaigns,” the blog warned, noting the need for skepticism of text just as “deepfakes” have taught skepticism of images.

Deepfakes are realistic, AI-generated synthetic videos or images of people doing or saying things they never actually did. By 2019, they had already sparked public concern. Based on the Generative Adversarial Network (GAN) framework introduced by Ian Goodfellow in 2014, these models pit a generator against a discriminator to produce increasingly convincing forgeries.^[72] Imagine a forger trying to paint counterfeit masterpieces (generator) while an art expert attempts to identify which ones are fake (discriminator). Each time the expert uncovers a fake, the forger learns and improves. Over time, this competition causes the fakes to become increasingly difficult to distinguish from genuine artworks. GANs quickly became associated with ethical risks, including nonconsensual pornography, misinformation, and manipulated media. “Deepfakes for text” served as a caution against convincing synthetic text produced at scale.

The decision to withhold the model was presented as a proactive “release strategy” intended to give the community time to discuss its implications. OpenAI acknowledged, “We are not sure that it is the right decision today,” but expressed hope that it would encourage more nuanced discussions about publication norms in AI, akin to debates in fields such as biotechnology and cybersecurity. In 2019, considering a model dual-use technology rather than an academic publication and emphasizing safety was unconventional. Unsurprisingly, the decision was met with skepticism and, at times, derision.

The media portrayed GPT-2 as a dangerous “fake news generator,” drawing public attention to both the capabilities and risks of AI. *Wired* ran a story titled “The AI Text Generator That’s Too Dangerous to Make Public” and quoted an OpenAI engineer who marveled at GPT-2’s realism: “It looks pretty darn real,” fearing it could enable “someone with malicious intent... to generate high-quality fake news.”^[73] *TechCrunch* echoed the story under the blunt title “OpenAI Built a Text Generator So Good, It’s Considered Too Dangerous to Release.”^[74]

However, many experts believed that the narrative overshadowed more nuanced discussions. Anima Anandkumar, an AI professor at Caltech and Director of Machine Learning Research at NVIDIA at the time, told *The Verge* that OpenAI’s approach was unnecessary from a safety standpoint and ultimately detrimental to the progress of AI research. Others in the field share this perspective, noting that open publication is especially important when risks are involved since transparency enables researchers to identify and develop safeguards.^[75]

Many joked that “OpenAI” had become “ClosedAI,” a remark on the organization’s shift away from its open-source roots. The “ClosedAI” meme was widely shared on Twitter and Reddit in 2019. [76] The community suspected OpenAI’s caution was a publicity move. Skeptics noted how the dramatic framing (“too dangerous to release”) generated media hype, benefiting OpenAI’s profile, and “accusations that OpenAI is motivated by hype-generation” emerged.[77] Although this controversy did attract attention, it was minuscule compared to the immense public focus OpenAI would soon receive from models like GPT-3 and ChatGPT.

Despite the criticism, OpenAI believed that a cautious approach was necessary. Policy researcher Miles Brundage emphasized that the objective wasn’t merely to focus on GPT-2 but to raise awareness about broader risks: “We tried to highlight both the current capabilities of GPT-2 and the risks of a broader class of systems... we should have been more precise on that distinction.” OpenAI clarified its stance in a blog post titled “GPT-2: 6-Month Follow-Up,” published on August 20, 2019. [78] The post outlined a staged release strategy, beginning with a 124 million parameter model in February, followed by a 355 million parameter model in May, and then releasing a 774 million parameter model along with the six-month follow-up.

This staggered approach soon influenced others in the field. In collaboration with the University of Washington, Hugging Face and the Allen Institute for Artificial Intelligence adopted a similar strategy for releasing their language model, Grover.[79],[80] In 2022, the Google Brain team withheld public release of its *Imagen* text-to-image diffusion model, citing ethical concerns.[81] In 2023, Meta restricted access to *LLaMA 1*, making it available only to approved academic researchers, government agencies, and civil society groups. The cautious precedent set by OpenAI prompted the industry to develop new norms and quickly became a template for navigating the tension between innovation and responsibility.

The six-month follow-up also included ongoing research with partners such as Cornell University, the Middlebury Institute of International Studies, the University of Oregon, and the University of Texas at Austin to evaluate the model’s potential for misuse and its societal impact. By the end of 2019, what had begun as a decision by one lab evolved into a community-wide dialogue on managing increasingly general-purpose AI systems that might be misused.

Looking back, the controversy surrounding GPT-2 stands out not because the model was especially good. It wasn’t, at least by today’s standards. GPT-2 wasn’t particularly dangerous either. Despite its content seeming convincingly human, the concern over its release turned out to be misplaced. OpenAI acknowledged as much when it released the full 1.5 billion parameter GPT-2 model in November 2019, noting, “We’ve seen no strong evidence of misuse so far.”[82]

And yet, OpenAI wasn’t entirely wrong, either. Traditional methods of detecting machine-generated content rely on analyzing text patterns such as repeated phrases or predictable linguistic structures. Yet, the primary concern with GPT-2 was the convincingly human-like text and its subtlety. GPT-2 generates contextually coherent outputs making it difficult to detect. So, GPT-2 was early anticipating the broader implications of increasingly capable AI models.

The significance of the GPT-2 moment may lie in its role as a harbinger of things to come. We now know that far more powerful models would soon emerge. Indeed, some were withdrawn shortly after public release. In November 2022, Meta introduced *Galactica*, a language model to assist scientists by synthesizing academic literature. It was taken down three days later amid concerns over its generation of authoritative-sounding misinformation that could erode public trust.[\[83\]](#) In 2024, Google followed a similar pattern, pulling *Gemini* only days after apologizing for its historically inaccurate outputs.[\[84\]](#) In retrospect, what once appeared premature caution with GPT-2 increasingly seems like a preview of the challenging decisions ahead.

GPT-2 was the first time the AI community publicly confronted the tension between innovation and acceleration on one hand and risk mitigation and safety on the other. Today, as language models write code, summarize meetings, and impersonate public figures, the GPT-2 moment feels almost quaint. However, it marked the first occasion in 75 years of research when anyone pressed pause, even though speculating about AI risk and control had long been a favorite pastime for researchers ranging from Turing and Wiener to Good and Minsky to Bostrom and Yudkowsky. These discussions largely remained theoretical, philosophical, or relegated to footnotes. Consequently, GPT-2 symbolizes a cultural moment indicating AI's first significant fracture. The tension between speed, scale, and safety transformed from merely abstract discussions to concrete concerns.

Yet, the GPT-2 decision did not relieve any tensions. In fact, by declaring the model "too dangerous to release," OpenAI opened Pandora's box. By publicly acknowledging risks that neither existed nor understood, OpenAI triggered a chain of events. This moment will precede the crisis that will unfold, leading to a slow march toward OpenAI's board firing CEO Sam Altman, citing vague safety concerns.

As previously noted, Ilya played a central role in the events leading to Altman's firing. Theories swirled that perhaps Sutskever had pushed out Altman due to discovering some dangerous capability or existential risk. The community latched onto the mystery, spawning countless posts and speculative threads.[\[85\]](#),[\[86\]](#) Many fixated on one haunting question: "What did Ilya see?" The meme became shorthand for something deeper that underscores the unspoken anxieties at the heart of AI. It captures a growing suspicion that those closest to the frontier might be witnesses to something the rest of us don't yet understand. Consequently, it elevates Sutskever's near-mythical image as a guardian at the edge. Combined with the lore around *Sutskever's List*, we get a picture of how the community views him: as someone who not only holds the keys to understanding modern AI but also might have a glimpse of its future risks. This dual reputation is echoed in memes like "Where's Ilya?" and "What did Ilya see?"[\[87\]](#)

1.4 Ovid's Unicorn

GPT-2 was introduced in the paper "Language Models are Unsupervised Multitask Learners," authored by Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever.[\[88\]](#) The media largely reacted with astonishment at the quality of GPT-2's generated text. The *Guardian* called GPT-2 a "revolutionary AI system" that "push[es] the boundaries of what was thought possible" in terms of output quality and versatility.[\[89\]](#)

The *Guardian* noted that GPT-2 can produce “*plausible*” continuations of prompts in many styles and domains, rarely exhibiting the grammatical nonsense or mid-sentence derailments that plagued previous systems. For example, after being seeded with the opening line of George Orwell’s Nineteen Eighty-Four: “It was a bright cold day in April, and the clocks were striking thirteen,” the system recognized the vaguely futuristic tone and the novelistic style and continued with: “I was in my car on my way to a new job in Seattle. I put the gas in, put the key in, and then I let it run. I just imagined what the day would be like. A hundred years from now. In 2045, I was a teacher in some school in a poor part of rural China. I started with Chinese history and history of science.” This lands somewhere between bleak futurism and a mediocre travel blog. The bizarre narrative leaps suggest the model is struggling to maintain narrative coherence rather than deliberately evoking Orwell’s unsettling style.

Nonetheless, GPT-2’s ability to generate coherent, though sometimes peculiar, text surprised and excited observers. Vox’s Kelsey Piper wrote that GPT-2 was “*one of the coolest AI systems I’ve ever seen*,” so human-like in writing that she quipped it “*may also be the one that will kick me out of my job.*”^[90] Reporters marveled at GPT-2’s zero-shot abilities. Without task-specific training, it could generate news articles from headlines or answer reading comprehension questions, achieving unimaginable results from an unsupervised model.^[91]

TechCrunch reported that GPT-2 produces “*longer text with greater coherence*” than prior models, calling it a vast improvement over the original GPT-1 model.^[92] Even OpenAI’s researchers were taken aback when GPT-2 produced a persuasive essay arguing a counterintuitive viewpoint, describing it as “something you could have submitted to the SAT and get a good score on.”^[93] Such anecdotes spread quickly, fueling excitement that AI had generated a new level of sophistication.

Researchers also voiced surprise. For instance, one researcher wrote that he was “pretty shocked” upon reading GPT-2’s now-famous “Ovid’s Unicorn” story, noting that “on the whole it’s remarkably coherent... like a news article that a human could have written”^[94] No prior model came close to GPT-2’s level of fluency, long-range coherence, and apparent world knowledge.

The model’s ability to generate multiple paragraphs of on-topic, grammatical text was an eye-opener for experts accustomed to earlier models that would quickly go off the rails. Some researchers highlighted that while GPT-2’s underlying techniques were not new, its scale made the difference. “*There is no algorithmic contribution here,... they are ‘just scaling up’ previous research, but seeing exactly how strong these scaled-up models are is an important contribution in its own right,*” one researcher noted, adding that “*if I had shown the [GPT-2] unicorn story to a colleague... and told them that it was AI-generated, I don’t think they would have believed me.*”^[95] The community recognized GPT-2 as proof that larger models can yield startling new performance levels.

GPT-2 is an attention-based artificial neural network known as a transformer with 1.5 billion parameters, trained on a dataset of approximately 8 million web pages and 40 GB of Internet text. GPT-2 is a direct scale-up of GPT, featuring over 10 times the parameters and trained on more than ten times the amount of data. It is trained with a simple objective: to predict the next word, given all previous words within some context. The diversity of the dataset causes this simple goal to contain naturally occurring demonstrations of many tasks across diverse domains.

The model generates synthetic text samples in response to being prompted with an arbitrary input. OpenAI described the model as "chameleon-like," meaning it adapts to the style and content of the conditioning text.[\[96\]](#) This allows the user to generate realistic and coherent continuations about a topic of their choosing, as seen by the following select sample taken directly from the GPT-2 paper and commonly known as "Ovid's Unicorn:"

System Prompt:

In a shocking finding, scientist discovered a herd of unicorns living in a remote, previously unexplored valley, in the Andes Mountains. Even more surprising to the researchers was the fact that the unicorns spoke perfect English.

GPT-2

Model Completion:

The scientist named the population, after their distinctive horn, Ovid's Unicorn. These four-horned, silver-white unicorns were previously unknown to science.

Now, after almost two centuries, the mystery of what sparked this odd phenomenon is finally solved.

Dr. Jorge Pérez, an evolutionary biologist from the University of La Paz, and several companions, were exploring the Andes Mountains when they found a small valley, with no other animals or humans. Pérez noticed that the valley had what appeared to be a natural fountain, surrounded by two peaks of rock and silver snow.

Pérez and the others then ventured further into the valley. "By the time we reached the top of one peak, the water looked blue, with some crystals on top," said Pérez.

Pérez and his friends were astonished to see the unicorn herd. These creatures could be seen from the air without having to move too much to see them – they were so close they could touch their horns.

While examining these bizarre creatures the scientists discovered that the creatures also spoke some fairly regular English. Pérez stated, "We can see, for example, that they have a common 'language,' something like a dialect or dialectic."

Dr. Pérez believes that the unicorns may have originated in Argentina, where the animals were believed to be descendants of a lost race of people who lived there before the arrival of humans in those parts of South America.

While their origins are still unclear, some believe that perhaps the creatures were created when a human and a unicorn met each other in a time before human civilization. According to Pérez, "In South America, such incidents seem to be quite common."

However, Pérez also pointed out that it is likely that the only way of knowing for sure if unicorns are indeed the descendants of a lost alien race is through DNA. "But they seem to be able to communicate in English quite well, which I believe is a sign of evolution, or at least a change in social organization," said the scientist.

Such an output may seem ordinary today. However, in 2019, it was shockingly good and nearly unimaginable two years earlier. Naive observers may have assumed that such a result relied on curated knowledge databases, including placenames, rules engines, or symbolic logic, to produce such coherent results. It's still hard to believe that a model trained to predict the next word could exhibit this level of narrative control and contextual awareness.

This sample is notable not only for its fluid and convincing prose but also for how effectively the model imitates the style and structure of an authentic news article. The initial sentence introduces the discovery of unicorns in the Andes, and the model dedicates ten sentences to elaborating on that single idea. It presents a fictional academic, Dr. Jorge Pérez, and places him at a plausible university, the University of La Paz, demonstrating sensitivity to geographical, academic, and stylistic contexts. It mirrors the conventions of journalism, such as switching from full name to surname in later references (i.e., "Pérez"), and incorporates plausible-sounding quotes and scientific speculation.

The model shows an impressive ability to track long-term dependencies connecting information across multiple paragraphs. This was impossible with earlier architectures. The model encoded real-world knowledge and effectively modeled statistical relationships between words. It approximates knowledge like:

- The Andes Mountains are in South America
- Jorge Pérez is a plausible name for a scientist in Bolivia
- Universities often bear city names like "La Paz"
- News articles introduce experts with titles and institutional affiliations

This implicit knowledge is not programmed into the model but is learned from the language used to describe the world. Predicting the next word requires modeling statistical relationships about the world that language refers to. As a result, without task-specific tuning, the model can generate stylistically nuanced, coherent, and context-sensitive text, even approximating implicit world knowledge.

As we follow the generated article, the model transitions to the second sentence of the prompt ("Even more surprising... the unicorns spoke perfect English") only after addressing the first in detail. This delayed yet coherent introduction of new information demonstrates the model's ability to structure its output like a human writer, capturing continuity and pacing. The story eventually ventures into fanciful territory, including theories of alien origins. However, the model was trained on Reddit, where whimsical ideas about alien origins are commonplace.

The model generated other nonsense too. The opening reference to "four-horned unicorns" exemplifies an oxymoron and highlights a fragile underlying "world model." In another sample, GPT-2 confidently describes "fires happening underwater."[\[97\]](#) These types of output became an early example of what we now call a hallucination, where models produce plausible-sounding text that is factually or logically incorrect. In 2019, OpenAI termed such outputs "failure modes."

It's important to note that "Ovid's Unicorn" was self-selected as the best among ten generated completions, which tempers any impression of consistency and reliability. However, even when generating nonsense, GPT-2's convincing form, structure, and tone changed expectations about machine-generated language. In other words, even a single high-quality sample like Ovid's Unicorn in 2019 was unprecedented, even if it was cherry-picked.

The GPT-2 paper provides fewer architectural and training details than the GPT-1 paper and, as noted, is not included on *Sutskever's List*. Therefore, we will skip an in-depth technical discussion and review GPT-2's performance across tasks as presented in the original paper. We'll explore how attention-based transformers like GPT-2 function in more detail in Chapter 5.

GPT-2 was trained on a dataset called WebText. As previously mentioned, WebText was created by scraping webpages linked from Reddit posts with at least three karma points, ensuring the content was generally informative, educational, or engaging rather than random or noisy. The final dataset included roughly 8 million documents and 40 GB of text. Wikipedia articles were excluded to prevent artificial performance boosts on standard NLP benchmarks.

The model acquired the ability to perform different language tasks without explicit supervision. The authors, including Ilya, referred to this phenomenon as "unsupervised multitask learning," where the model's diverse capabilities emerge from training it solely to predict text continuations. This approach differed from previous specialized systems, which relied heavily on carefully labeled datasets and task-specific training.

The researchers tested various models with increasing sizes, ranging from 117 million parameters similar to the original GPT model to 1.5 billion parameters. Larger models outperformed smaller ones across various tasks, demonstrating a consistent log-linear relationship between scale and performance. The largest GPT-2 model achieved state-of-the-art zero-shot results, meaning it performed tasks without prior training or examples, on seven out of eight standard language modeling benchmarks.

However, performance across more specialized benchmarks revealed limitations. When researchers prompted GPT-2 with the phrase "TL;DR:" (meaning "too long; didn't read") to generate summaries without explicit training, the results qualitatively resembled standard summaries but quantitatively scored only marginally better than randomly selected sentences, and significantly below advanced summarization techniques.

Similarly, GPT-2 struggled in multilingual tasks without explicit translation training (i.e., "zero-shot"). The model produced poor translations, scoring a 5 out of 100 for English-to-French and 11.5 out of 100 for French-to-English on the WMT-14 dataset. In open-domain question-answering, GPT-2 answered about 4.1% of questions correctly, indicating limited factual knowledge. Yet, on a commonsense reasoning test known as the Winograd Schema Challenge, GPT-2 achieved 70.7% accuracy, surpassing prior state-of-the-art methods by roughly 7%.

GPT-2 was impressive but not powerful enough to pose a threat. The staged rollout revealed these fears to be overstated. To most outside observers, neither its benchmark results nor demonstrations like "Ovid's Unicorn" represented a transformative breakthrough. Yet, to researchers involved, particularly Ilya and colleagues, GPT-2 marked a milestone. It underscored how unsupervised learning and scaling quietly paved the way toward larger, more powerful models.

In an October 2019 interview with *The New Yorker*, Sutskever marveled at GPT-2's unexpected prowess, saying, "Give it the compute, give it the data, and it will do amazing things... it's like alchemy!"^[98] The reporter captured him with eyes "wide with wonder," underscoring how astonishing the model's capabilities appeared, even to an expert who understood the underlying mechanics. For Sutskever, the metaphor of alchemy highlighted how scaling compute and data had produced outcomes that seemed almost miraculous.

In late November 2019, *The Economist* published an interview conducted entirely with GPT-2 as an experimental curiosity. Sutskever shared this article on Twitter (now X), noting enthusiastically: "The Economist interviews GPT-2—and the interview makes sense."^{[99],[100]} This public remark illustrated his delight at the model's capacity to produce coherent, topical responses indistinguishable at times from human-generated text.

Sutskever suggests that GPT-2 was doing more than merely stringing together plausible sentences; it was starting to show real understanding. In a May 2020 appearance on the Lex Fridman podcast (May 2020; Episode 94), he articulated this clearly, explaining that larger language models like GPT-2 demonstrated unmistakable "signs of semantic understanding" absent in smaller models.^[101] Though he clarified that this understanding was not yet complete—a characteristic restraint often found in his commentary—he emphasized that GPT-2 undeniably understood language meaning, at least in part.^[102]

Ultimately, GPT-2 mattered on multiple levels, including technically, culturally, and philosophically, though its immediate technical impact never matched its sweeping philosophical conclusions. Declaring GPT-2 "too dangerous to release" shifted the Overton window, bringing once-fringe concerns about AI safety into mainstream discourse. While the risks never materialized, the event set a precedent for preemptive caution, establishing a new normal. This tension between AI's capabilities, perceived threats, and the responsibilities accompanying scale would erupt dramatically four years later with Sam Altman's firing, underscoring GPT-2's lasting significance as both a cautionary tale and a cultural turning point.

- [1] <https://x.com/elonmusk/status/1727960652390547600>
- [2] <https://mashable.com/article/openair-ilya-slutskever-leaves-chief-scientist>
- [3] <https://twitter.com/elonmusk/status/1734063208648880247>
- [4] <https://dallasinnovates.com/exclusive-qa-john-carmacks-different-path-to-artificial-general-intelligence/>
- [5] <https://www.utoronto.ca/news/godfather-ai-geoffrey-hinton-s-nobel-prize-draws-headlines-around-world>
- [6] https://proceedings.neurips.cc/paper_files/paper/2012/file/c399862d3b9d6b76c8436e924a68c45b-Paper.pdf
- [7] (pg. 4) <https://web.cs.ucla.edu/~kaoru/FordMR2018Architects-of-Intelligence.pdf#:~:text=advance%20of%20computing%20power%2C%20together,a%20nearly%20incomprehensible%20increase%20in>
- [8] "How a stubborn computer scientist accidentally launched the deep learning boom". Ars Technica. 11 November 2024. Retrieved 24 March 2025.
- [9] <https://computerhistory.org/blog/chm-releases-alexnet-source-code/>
- [10] For details: <https://www.nytimes.com/2023/12/03/technology/ai-openai-musk-page-altman.html>

- [11] <https://www.businessinsider.com/the-surprising-backgrounds-of-people-working-at-google-brain-2016-8>
- [12] <https://arxiv.org/pdf/1603.04467>
- [13] https://proceedings.neurips.cc/paper_files/paper/2012/file/6aca97005c68f1206823815f66102863-Paper.pdf
- [14] Silver, D., Huang, A., Maddison, C. et al. Mastering the game of Go with deep neural networks and tree search. *Nature* 529, 484–489 (2016). <https://doi.org/10.1038/nature16961>
- [15] <https://arxiv.org/abs/1409.3215>
- [16] In an interview with CNBC, Musk referred to Sutskever as the "linchpin" of OpenAI's success. <https://www.wired.com/story/what-openai-really-wants/>
- [17] <https://www.nytimes.com/2023/12/03/technology/ai-openai-musk-page-altman.html>
- [18] <https://youtu.be/2BfMuHdfGJ?si=HPLwKLgYwvgzl8K&t=2772>
- [19] Season 1, Episode 22 https://youtu.be/fCoavgGZ64Y?si=R_tfONE0OCdShCEk&t=1780
- [20] <https://www.wired.com/story/what-openai-really-wants/>
- [21] <https://youtu.be/bWr-DA5Wjfw?si=PKSg9nuLEyWhVKHU&t=203>
- [22] <https://www.businessinsider.com/meet-openai-cofounder-ilya-sutskever-supported-regretted-sam-altman-firing-2023-11#at-the-end-of-2015-ilya-sutskever-cofounded-open-ai-with-ex-ceo-sam-altman-7>
- [23] <https://www.wired.com/story/what-openai-really-wants/>
- [24] <https://web.archive.org/web/20211006105333/https://openai.com/blog/introducing-openai/>
- [25] <https://www.wired.com/story/what-openai-really-wants/>
- [26] <https://openai.com/index/openai-gym-beta/>
- [27] <https://openai.com/index/dota-2/>
- [28] <https://www.cnbc.com/2018/02/21/elon-musk-is-leaving-the-board-of-openai.html>
- [29] <https://youtu.be/2BfMuHdfGJ?si=HPLwKLgYwvgzl8K&t=2772>
- [30] <https://www.businessinsider.com/elon-musk-reportedly-tried-lead-openai-left-after-founders-objected-2023-3>
- [31] <https://www.theguardian.com/technology/2019/feb/14/elon-musk-backed-ai-writes-convincing-news-fiction>
- [32] Notably in the 2020 paper, "Language Models are Few-Shot Learners"
- [33] <https://analyticsindiamag.com/the-brain-that-supercharged-chatgpt-imagenet-tensorflow/>

- [34] <https://www.businessinsider.com/meet-openai-cofounder-ilya-sutskever-supported-regretted-sam-altman-firing-2023-11#sutskever-said-he-spends-the-bulk-of-his-days-working-9>
- [35] <https://x.com/ilyasut/status/1491554478243258368>
- [36] <https://x.com/karpathy/status/1491608794253512707>
- [37] <https://x.com/ylecun/status/1492604977260412928>
- [38] <https://x.com/mpshanahan/status/1491715721289678848>
- [39] <https://fortune.com/2024/05/21/openai-superalignment-20-compute-commitment-never-fulfilled-sutskever-leike-altman-brockman-murati/>
- [40] <https://openai.com/index/introducing-superalignment/>
- [41] <https://futurism.com/openai-employees-say-firms-chief-scientist-has-been-making-strange-spiritual-claims>
- [42] <https://www.theatlantic.com/technology/archive/2023/11/sam-altman-open-ai-chatgpt-chaos/676050/>
- [43] <https://www.axios.com/2023/11/20/sam-altman-fired-openai-board-illya-sutsever-regrets>
- [44] <https://futurism.com/openai-employees-say-firms-chief-scientist-has-been-making-strange-spiritual-claims>
- [45] <https://www.reuters.com/technology/artificial-intelligence/openai-co-founder-sutskevers-new-safety-focused-ai-startup-ssi-raises-1-billion-2024-09-04/#:~:text=Last%20year%2C%20he%20was%20a,breakdown%20of%20communications>
- [46] <https://www.reuters.com/technology/sam-altmans-ouster-openai-was-precipitated-by-letter-board-about-ai-breakthrough-2023-11-22/#:~:text=Nov%202022%28Reuters%29%20,with%20the%20matter%20told%20Reuters>
- [47] <https://www.reuters.com/technology/sam-altmans-ouster-openai-was-precipitated-by-letter-board-about-ai-breakthrough-2023-11-22/#:~:text=Some%20at%20OpenAI%20believe%20Q,in%20most%20economically%20valuable%20tasks>
- [48] <https://www.reuters.com/technology/sam-altmans-ouster-openai-was-precipitated-by-letter-board-about-ai-breakthrough-2023-11-22/#:~:text=Nov%202022%28Reuters%29%20,with%20the%20matter%20told%20Reuters>
- [49] <https://mashable.com/article/openair-ilya-slutskever-leaves-chief-scientist#:~:text=That%20something%20was%20amiss%20was,and%20Altman%27s%20posts%20on%20X>
- [50] <https://twitter.com/elonmusk/status/1734063208648880247>
- [51] <https://arstechnica.com/information-technology/2023/11/report-sutskever-led-board-coup-at-openai-that-ousted-altman-over-ai-safety-concerns/>

- [52] <https://www.reuters.com/technology/sam-altmans-ouster-openai-was-precipitated-by-letter-board-about-ai-breakthrough-2023-11-22#:~:text=The%20previously%20unreported%20letter%20and,solidarity%20with%20their%20fired%20leader>
- [53] <https://x.com/ilyasut/status/1726590052392956028>
- [54] <https://x.com/ilyasut/status/1726590052392956028>
- [55] <https://x.com/OpenAI/status/1727206187077370115?s=20>
- [56] <https://fortune.com/2024/05/21/openai-superalignment-20-compute-commitment-never-fulfilled-sutskever-leike-altman-brockman-murati/>
- [57] <https://chatgpt.com/c/67e88bf5-374c-8002-9de3-77f74ca68d99>
- [58] <https://futurism.com/the-byte/ilya-sutskever-leaves-openai>
- [59] <https://www.vox.com/future-perfect/2024/5/17/24158403/openai-resignations-ai-safety-ilya-sutskever-jan-leike-artificial-intelligence>
- [60] <https://x.com/janleike/status/1791498174659715494>
- [61] <https://www.reuters.com/technology/artificial-intelligence/openai-co-founder-sutskevers-new-safety-focused-ai-startup-ssi-raises-1-billion-2024-09-04>
- [62] https://dallasinnovates.com/exclusive-qa-john-carmacks-different-path-to-artificial-general-intelligence?utm_source=www.turingpost.com&utm_medium=referral&utm_campaign=the-mysterious-ai-reading-list-ilya-sutskever-s-recommendations
- [63] <https://www.reuters.com/technology/chatgpt-sets-record-fastest-growing-user-base-analyst-note-2023-02-01/>
- [64] https://news.ycombinator.com/item?id=34641359&utm_source=www.turingpost.com&utm_medium=referral&utm_campaign=the-mysterious-ai-reading-list-ilya-sutskever-s-recommendations
- [65] https://x.com/ID AA Carmack/status/1622673143469858816?utm_source=www.turingpost.com&utm_medium=referral&utm_campaign=the-mysterious-ai-reading-list-ilya-sutskever-s-recommendations
- [66] <https://github.com/dzyim/ilya-sutskever-recommended-reading?tab=readme-ov-file>
- [67] <https://x.com/keshavchan/status/1787861946173186062>
- [68] <https://www.technologyreview.com/2023/10/26/1082398/exclusive-ilya-sutskever-openais-chief-scientist-on-his-hopes-and-fears-for-the-future-of-ai/>
- [69] <https://www.turingpost.com/p/ilya-sutskever-reading-list>
- [70] <https://www.theguardian.com/technology/2019/feb/14/elon-musk-backed-ai-writes-convincing-news-fiction#:~:text>New%20AI%20fake%20text%20generator,dangerous%20to%20release%2C%20say%20creators>

- [71] <https://openai.com/index/better-language-models/>
- [72] <https://arxiv.org/abs/1406.2661>
- [73] <https://www.wired.com/story/ai-text-generator-too-dangerous-to-make-public/>
- [74] <https://techcrunch.com/2019/02/17/openai-text-generator-dangerous/>
- [75] <https://thegradient.pub/openai-please-open-source-your-language-model/>
- [76] <https://thegradient.pub/openai-please-open-source-your-language-model/>
- [77] <https://medium.com/data-science/openais-gpt-2-the-model-the-hype-and-the-controversy-1109f4bfd5e8>
- [78] <https://openai.com/index/gpt-2-6-month-follow-up/>
- [79] <https://medium.com/huggingface/ethical-analysis-of-the-open-sourcing-of-a-state-of-the-art-conversational-ai-852113c324b2>
- [80] <https://thegradient.pub/why-we-released-grover/>
- [81] <https://Imagen.research.google>
- [82] <https://openai.com/index/gpt-2-1-5b-release/>
- [83] <https://www.technologyreview.com/2022/11/18/1063487/meta-large-language-model-ai-only-survived-three-days-gpt-3-science/>
- [84] <https://apnews.com/article/google-gemini-ai-chatbot-image-generation-1bd45f1e67dfe0f88e5419a6efe3e06f>
- [85] <https://twitter.com/elonmusk/status/1734063208648880247>
- [86] <https://mashable.com/article/openair-ilya-slutskever-leaves-chief-scientist#:~:text=That%20something%20was%20amiss%20was, and%20Altman%27s%20posts%20on%20X>
- [87] <https://www.theverge.com/2024/5/14/24156920/openai-chief-scientist-ilya-sutskever-leaves>
- [88] https://cdn.openai.com/better-language-models/language_models_are_unsupervised_multitask_learners.pdf
- [89] <https://www.theguardian.com/technology/2019/feb/14/elon-musk-backed-ai-writes-convincing-news-fiction>
- [90] <https://www.vox.com/future-perfect/2019/2/14/18222270/artificial-intelligence-open-ai-natural-language-processing>
- [91] <https://www.theverge.com/2019/2/14/18224704/ai-machine-learning-language-models-read-write-openai-gpt2>
- [92] <https://techcrunch.com/2019/02/17/openai-text-generator-dangerous>

- [93] <https://www.theverge.com/2019/2/14/18224704/ai-machine-learning-language-models-read-write-openai-gpt2>
- [94] <https://medium.com/data-science/openais-gpt-2-the-model-the-hype-and-the-controversy-1109f4bfd5e8>
- [95] <https://medium.com/data-science/openais-gpt-2-the-model-the-hype-and-the-controversy-1109f4bfd5e8>
- [96] <https://openai.com/index/better-language-models/>
- [97] <https://openai.com/index/better-language-models/>
- [98] <https://www.newyorker.com/magazine/2019/10/14/can-a-machine-learn-to-write-for-the-new-yorker>
- [99] <https://x.com/ilyasut/status/1199036860934193152>
- [100] <https://medium.economist.com/how-i-sort-of-interviewed-an-artificial-intelligence-2a9c069a1680>
- [101] <https://www.happyscribe.com/public/lex-fridman-podcast-artificial-intelligence-ai/94-ilya-sutskever-deep-learning>
- [102] 1:00:36 <https://www.happyscribe.com/public/lex-fridman-podcast-artificial-intelligence-ai/94-ilya-sutskever-deep-learning>

2 The AlexNet Moment

This chapter covers

- Skepticism about artificial neural networks
- Feature engineering before AlexNet
- Training artificial neural networks is hard
- ImageNet's role in AlexNet's success
- The AlexNet Moment and technical innovations

Papers

- ImageNet Classification with Deep Convolutional Neural Networks
(Krizhevsky, Sutskever, & Hinton, 2012)

In 2012, Ilya, alongside Alex Krizhevsky and Geoffrey Hinton, trained a convolutional neural network to classify images. Their network stunned the AI community by dramatically reducing error rates in the emerging ImageNet Large Scale Visual Recognition Challenge (ILSVRC). AlexNet achieved a top-5 error rate of 15%, significantly outperforming conventional methods reliant on handcrafted feature engineering, which produced around 26% error. Top-5 error rate measures how often the correct answer is not among the model's top five guesses.

Although the original paper did not explicitly name the model, the community adopted “AlexNet,” following an informal tradition of naming influential networks after their lead authors, as previously seen with *Hopfield Networks* after John Hopfield, *LeNet* after Yann LeCun, and *DanNet* after Dan Ciresan.[\[1\]](#)[\[2\]](#)[\[3\]](#) Later models like VGGNet from the Visual Geometry Group at the University of Oxford and ResNet (Chapter 4) would shift toward more descriptive or brand-like names, often chosen directly by their authors. Nevertheless, after years of false starts, AlexNet demonstrated that artificial neural networks were remarkably effective at pattern recognition.[\[4\]](#)

The AlexNet paper has become highly cited. By spring 2025, it had been cited over 174,000 times. While some reports cautioned that these models remained fragile, requiring substantial data and sometimes making peculiar errors, the overall tone during that period was celebratory. The revolution, it seemed, had arrived. AlexNet transformed artificial neural networks from speculative curiosity into a strategic investment, marking a fundamental shift across academia, industry, and public perception. It is not an overstatement to say that this moment turned deep learning from a niche interest into a global phenomenon.

Yet, to feel the full weight of the AlexNet moment, we must understand the world it entered: the assumptions it defied, the failures that preceded it, and the quiet persistence of those who believed it could work.

Today, artificial neural networks are taken for granted, yet their history had been characterized by brief moments of enthusiasm and long periods of disillusionment. A revival began in 1980s when David Rumelhart, Geoffrey Hinton, and Ronald Williams popularized backpropagation, which provided a practical algorithm for training multi-layer artificial neural networks.[\[5\]](#) Backpropagation computes the gradient of the loss with respect to each weight in a network. This allows the network to adjust its parameters in a way to reduce errors during training. This breakthrough renewed optimism in connectionist (i.e., artificial neural networks) approaches and helped lay the foundation for modern deep learning.

A few years later, LeNet-5, named for its five-layer architecture, demonstrated the potential of a specialized artificial neural network called a convolutional neural network (CNNs or ConvNets). The impressive performance in handwritten digit recognition indicated that CNNs could scale to real-world tasks. However, that promise did not fully materialize. Training deeper networks was challenging, large datasets were scarce, and computing power remained limited. While connectionist techniques had shown promise on small problems, they did not generalize, and momentum faded. By the late 1990s, artificial neural networks (ANNs) were seen as a dead end. One observer recalled that “[artificial] neural networks had become a backwater” and “many researchers had moved on.”[\[6\]](#) Indeed, throughout the 1990s and 2000s, ANNs were “almost entirely dismissed as researchers embraced other approaches.”[\[7\]](#)

The academic community organized around methods with a solid theoretical foundation and good performance on limited data. Support Vector Machines (SVMs) gained popularity due to convex optimization, an appealing mathematical property ensuring globally optimal solutions could be found efficiently.[\[8\]](#) AdaBoost showcased how ensembles of simple learners could outperform deeper networks in tasks such as face detection.[\[9\]](#)[\[10\]](#) In contrast to SVMs, ANNs lack theoretical guarantees. They were difficult to analyze and were correctly perceived as slow black boxes. Training a multi-layer network on high-dimensional data could take days, often leading to problems like exploding or vanishing gradients. Moreover, these networks were prone to overfitting without large datasets, which were nearly impossible to obtain. It also tended to get stuck in poor local minima: points where the error is low, but not the lowest possible, preventing further improvement. Lacking today's data, compute resources, and architectural enhancements, artificial neural networks of the era failed to deliver the promised gains, prompting many researchers to abandon them.[\[11\]](#) As a 2015 deep learning review by Geoffrey Hinton, Yann LeCun, and Yoshua Bengio noted, even ConvNets, which had some success in the 90s, were "forsaken by the mainstream computer vision and machine learning communities until the ImageNet competition in 2012."[\[12\]](#)

Only a small group of believers kept the research alive throughout this period. Hinton, LeCun, Bengio, and a handful of others continued to develop ideas during the 2000s. Their efforts, including Hinton's research on deep belief nets in 2006 and Bengio's greedy layer-wise training in 2007, were acknowledged but failed to persuade the broader academic community, which remained skeptical.[\[13\]](#)[\[14\]](#)[\[15\]](#) The prevailing wisdom favored more practical methods. In hindsight, these researchers sustained the field during the AI winter. Yet AlexNet's crucial optimization insights finally bridged these dormant theoretical ideas to practical breakthroughs.

2.1 Feature Engineering vs. Representation Learning

To appreciate the significance of AlexNet's breakthrough, we must understand the dominant paradigm it disrupted. During the 2000s, progress in computer vision was dominated by hand-crafted feature engineering. The task required designing robust feature descriptors by hand rather than learning feature representations directly from data without human direction. Classic feature descriptors such as SIFT (Scale-Invariant Feature Transform) and SURF (Speeded-Up Robust Features) were extensively used to detect and describe local image patches.[\[16\]](#)[\[17\]](#)[\[18\]](#) Global image characteristics were encoded using bag-of-visual-words, histograms of quantized SIFT descriptors, and spatial pyramids.[\[19\]](#) Researchers would pipeline these engineered features into a classifier like an SVM for image classification. This approach yielded state-of-the-art results for years.

The HOG (Histogram of Oriented Gradients) descriptor exemplifies the mindset of the pre-deep learning era.[\[20\]](#) As one paper observed, "HOG features were the standard, off-the-shelf image feature representation before the 2012 advent of powerful CNNs."[\[21\]](#) When combined with SVM, HOG achieved strong performance in tasks like pedestrian detecting.[\[22\]](#) By the late 2000s, this approach was refined into the Deformable Parts Model (DPM), which used HOG features and a latent structured SVM to represent objects as collections of parts with learnable spatial relationships.[\[23\]](#) DPM became a leading method on the PASCAL Visual Object Challenge (VOC) benchmarks for object detection.[\[24\]](#)

Most high-performing vision systems of that era relied on hand-crafted hierarchical feature pipelines. Local descriptors such as SIFT and SURF were used to detect and describe image patches, while bag-of-visual-words models, histograms of quantized descriptors, and spatial pyramids encoded global representations. These feature extraction methods were paired with shallow learning algorithms like SVMs and boosted decision trees that mapped input features directly to outputs without learning intermediate or hierarchical representations.

Between 2007 and 2012, the dominant architecture for object recognition remained remarkably stable. Researchers relied on local descriptors such as SIFT, combined with bag-of-visual-words, spatial pyramid matching, and SVM classifiers.[\[25\]](#) Even as late as 2012, the winning entry in the PASCAL VOC used this approach. Innovation during this period focused on refining these pipelines through improved pooling strategies, kernel tricks, contextual re-scoring, and subcategory modeling.[\[26\]](#)[\[27\]](#) Architectures that deviated from this formula failed to outperform these well-tuned baselines and risked overfitting the relatively small PASCAL dataset. Despite progress in other areas of machine learning, the field of visual recognition remained closely linked to hand-crafted features and carefully engineered data pipelines.

A telling indicator of the field's mood in the 2000s can be seen at major computer vision conferences such as CVPR, ICCV, and PASCAL VOC. These venues featured almost no papers on artificial neural networks. Instead, they were dominated by research on feature descriptors, segmentation algorithms, and shallow kernel methods like SVMs. During this period, publishing anything involving ANNs was difficult and advocating for them was a contrarian position.

Computer vision researchers preferred hand-crafted features because they incorporated prior knowledge about visual invariances. For example, SIFT was designed to manage variations in scale and rotation, while HOG effectively captured edge patterns related to shape detection. These features were considered reliable, interpretable, and effective especially when training data was limited. In contrast, automatic feature learning with neural networks was risky, opaque, and dependent on large datasets. The prevailing belief was that effective features *needed* to be created by hand. Moreover, many thought the image signal was too noisy and high-dimensional for a general-purpose learning algorithm to identify robust visual features. A few contrarians argued that features should be learned directly from data, yet all failed. The field placed more trust in human intuition than in data; believing features were better designed than discovered.

While learned representations could, at least in principle, adapt more easily to changes in data distribution, the argument remained theoretical, as no learned features could match the performance of methods like HOG or SIFT. Yet, traditional methods had weaknesses, such as sensitivity to minor input perturbations and a few people were noticing, including Ilya Sutskever. [28] For instance, a HOG-based detector may react strongly to irrelevant edge patterns or struggle under minor image distortions like detecting a “car” in a patch of water that a learned CNN feature could better manage.[29] See Figure 2.1. These weaknesses became increasingly clear as more researchers recognized that hand-crafted methods like HOG and SIFT relied on static, manually engineered pipelines heavily tailored to the specific datasets, like PASCAL, used for evaluation. These benchmarks introduced biases and constraints that inadvertently shaped research priorities. In 2011, Alexei “Alyosha” Efros and Antonio Torralba highlighted such concerns, arguing that PASCAL fostered overfitting, inhibited generalization, and encouraged models that excelled only in constrained, “closed world” settings, thus casting doubt on the robustness of even the best hand-crafted pipelines.[30][31]

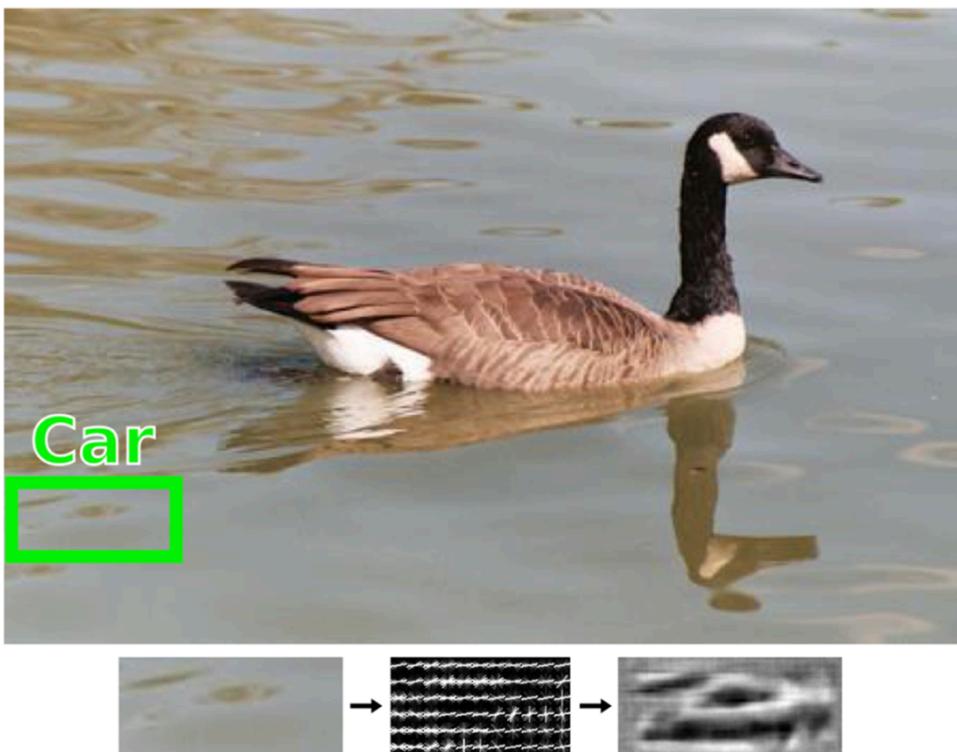


Figure 2.1 The figure shows an image of a goose swimming (top), where an object detector mistakenly labels a small patch of rippled water as a car. Below this image are three visualizations that clarify the source of this error: a close-up of the water patch (left), the corresponding HOG features (middle), and a specialized visualization (right) demonstrating that the rippled water's features resemble those associated with cars in HOG feature space. *Vondrick et al. (2013)*, **Figure 2.1**. Used with explicit permission granted by the lead author (Carl Vondrick).

In other words, HOG and SIFT appeared robust because they were optimized for specific benchmarks like PASCAL, masking their poor generalization to the broader and messier reality of open-world vision problems. Computer vision had become benchmark-driven and dataset-bound, optimized for benchmark performance within the contrived Visual Object Challenge (VOC), instead of real-world generalization. In short, being “dataset-bound” implies that the vision community, whether knowingly or not, was optimizing for the success of its own choosing, which included artificial constraints of rigid features and specific benchmarks. Efros and Torralba exposed a blind spot to the entire field.

Nonetheless, standard practice in the 2000s did not focus on expanding data size. Instead, the emphasis remained on clever architectures and improved hand-tuning of pipelines within fixed datasets. Progress was measured in marginal gains on narrow benchmarks, and scaling models and data remained foreign to nearly everyone. Multi-layer artificial neural networks were considered elegant in theory but too brittle, computationally expensive, and data-hungry to be practical. The breakthroughs that would eventually validate their promise still lay ahead, awaiting not only algorithmic innovations but also a fundamental shift in mindset: designing features to learn them and optimizing within datasets to questioning the datasets themselves.

2.2 Pre-AlexNet Skepticism

During this era, many leading figures in machine learning doubted that artificial neural networks could rival established methods. Jitendra Malik, a prominent computer vision researcher at the University of California, Berkeley, exemplified the academic skepticism that defined the field. His stance reflected a broader “show-me” attitude, where researchers demanded clear empirical proof. Aware of this skepticism, Geoffrey Hinton approached Malik in 2011 and asked what kind of evidence would be convincing. Malik replied, “You really want to impress me? Show me they can handle the PASCAL VOC object recognition challenge.”

Malik’s reply reflects a philosophical stance similar to Russell’s Teapot: the burden of proof lies with the person making a claim rather than shifting the burden of disproof to others. Malik’s skepticism was reasonable at the time. Artificial neural networks had not yet outperformed the best hand-crafted feature pipelines on any respected benchmark. The vision community was waiting for evidence strong enough to overcome well-founded doubts. While this is a healthy posture in any scientific discipline, the tone of the exchange may also reveal a field that had been burned before.

Many felt we tried artificial neural networks in the ‘90s, but they didn’t work. Why would they work now? Past failures loomed large, and disappointments hardened a widespread sentiment. The community hadn’t forgotten that the “second wave” of artificial neural networks that began in the 1980s, and “ran out of steam” by the late 90s.^[32] Younger researchers inherited this skepticism, often seeing artificial neural nets as anachronistic and unorthodox.^[33] By 2011, the field had become incredulous, adopting a “show me, not tell me” stance. No amount of theoretical promise or anecdotal success would be enough; researchers were demanding to be overwhelmed, but not by improvements driven by data alone.

One researcher who clearly articulated this skepticism was Alyosha Efros, who argued that many apparent breakthroughs in artificial neural networks were driven more by larger datasets than genuine algorithmic innovation. Like many at the time, Efros was unconvinced that neural networks benefited from scale in a way fundamentally different from simpler models. He believed that if given enough data, a simpler method could perform just as well. In effect, his critique suggests that connectionist researchers were attributing too much success to their architectures and not enough to their training data.[\[34\]](#)

Efros recalls reviewing a 2008 paper in which an artificial neural network outperformed his earlier work on image localization. Suspecting the gain came from larger datasets, he asked the authors to evaluate a simple nearest neighbor method with the same dataset. They found that the artificial neural network performed no better than the simple nearest neighbor. "If you keep the data size constant, the fancy schmancy [artificial] neural network was doing no better than a simple nearest neighbor," Efros said. He added, "I'm not saying 'forget [artificial] neural networks,... but in this particular setting, it was really data that was doing all the main lifting."[\[35\]](#) This stance gained traction as the field started to focus on larger datasets. The emerging rise of "big data" lent additional weight to the idea that scale, not architecture, was the primary engine of progress.

In 2009, Alon Halevy, Peter Norvig, and Fernando Pereira at Google wrote "*The Unreasonable Effectiveness of Data*" (cited over 2,500 times), arguing that more data yielded better model performance.[\[36\]](#) This position aligns with Efros' critique that "fancy schmancy" architectures without sufficient data are little more than elaborate scaffolding, and even simple methods like nearest neighbors become comparably effective. Thus, the paper positioned data as a universal equalizer of sorts, suggesting that all statistical approaches, but especially simple ones, benefit from increased scale. Their paper made no mention of artificial neural networks. During this time, even Google saw the futility of working on anything besides n-grams, which they mention seven times.[\[37\]](#)

The lesson for critics was clear: artificial neural networks weren't alchemical. They didn't conjure intelligence from structure alone or uniquely benefit from more data. In fact, they were widely considered a liability. At best, they reflected what the data contained, leading many to conclude that any method with more data might suffice. Both "*The Unreasonable Effectiveness of Data*" and Efros' broader critique argue that researchers should first amass significantly more data; without it, all models, even artificial neural networks, remain "stone soup," as Efros frequently notes in his talks.[\[38\]](#)

In the folktale Stone Soup, hungry travelers arrive in a village where the locals insist they have no food to spare. Undeterred, the travelers begin making "stone soup" with only water and stones, expressing how much better it would taste with various ingredients. Intrigued and inspired, the villagers each contribute food items like carrots, onions, barley, buttermilk, and even a chicken. The pot transforms into a delightful communal meal. Eventually, the villagers marvel at how such a wonderful soup emerged from just a few stones.

A modern retelling might feature researchers who promise to create artificial intelligence. The researchers suggest it would perform optimally with all their data, images, and text, and even better with human feedback, allowing users to contribute their efforts. Researchers advocate for prompt engineering, and users readily comply. Ultimately, users marvel at the machine's intelligence, believing it was constructed from a few simple algorithms while overlooking the fact that they provided many of the essential elements.[\[39\]](#)

There is a lot of truth in this modern retelling. The field needed more ingredients, and data wasn't getting enough credit. Yet, unlike the tale's universal recipe, not all algorithms benefit equally. While the belief that more data is better was rightfully gaining momentum, the idea that data alone acts as a great equalizer masks a deeper truth: certain architectures derive disproportionately greater value from scale. ImageNet would soon provide abundant data, but rather than democratizing the field, it would crown a single champion, and it was not SIFT+SVM or nearest neighbors.

2.3 ImageNet

Jitendra Malik's challenge to Hinton was explicit: "Show me they can handle the PASCAL VOC object recognition challenge."^[40] PASCAL VOC was a modestly sized but meticulously annotated dataset that also served as a fierce proving ground for computer vision algorithms.^[41] For Malik, VOC was the serious benchmark that could separate hype from substance. "Show me they [artificial neural networks] can handle something serious," Malik added. Hinton responded, "No luck... It's just too small." Hinton pressed for an alternative. Malik pointed Hinton to the ImageNet project and its upcoming competition. Malik told Hinton: "Fine, if not PASCAL, then go win ImageNet." If artificial neural networks were to surpass the hand-crafted paradigm, this would be where it prove it.^[42] The gauntlet was thrown.

AI researcher Fei-Fei Li began developing ImageNet in 2006 at Princeton. The project was inspired by cognitive psychologist Irving Biederman, who sought to explain the remarkable efficiency of human visual recognition. Biederman proposed that recognizable objects are composed of a limited set of basic shapes, or "Geons."^[43] Identifying 36 such Geons, he suggested humans perceive roughly 30,000 meaningful combinations of shapes known as the "Biederman number." This convergence between cognitive theory and large-scale visual categorization provided a conceptual foundation for ImageNet.^[44] The ImageNet dataset and accompanying paper were released in 2009, with the first major competition taking place the following year, in 2010.

Eleven teams participated in the first competition, with the winning entry using HOG features combined with an SVM. The 2011 competition saw fewer participants and was again won by an SVM-based approach. Fei-Fei Li had bet that more data would yield better performance, but the early results were underwhelming. Traditional techniques dominated, delivering incremental improvements rather than breakthroughs. Researchers were still crafting features by hand, and the scale of ImageNet offered diminishing returns without an architecture capable of learning representations from the size and diversity of the data. ImageNet risked becoming a beautiful but useless monument to scale.

Yet, Hinton recognized that ImageNet was the perfect battleground, large enough to allow a multi-layer artificial neural network to learn feature representations and a benchmark the community was beginning to care about. In 2012, ImageNet was smaller in absolute terms than it is today. Currently, ImageNet contains more than 14 million labeled images across more than 22,000 categories, converging on the Biederman number. However, in relative terms, ImageNet was enormous. Upon its release in 2009, the dataset featured 3.2 million images and over 5,000 categories. In 2012, the ImageNet-1k, a widely used subset utilized during the ImageNet Large Scale Visual Recognition Challenge (ILSVRC) from 2012 to 2017, contained approximately 1.2 million images across 1,000 categories. For comparison, in 2012, PASCAL VOC included 20 categories and approximately 11,000 images.[\[45\]](#)

Fei-Fei Li built ImageNet to push the field of computer vision and the limits of visual recognition algorithms through scale. Li's ambitious endeavor to create the ImageNet dataset was also met with skepticism from peers, including Malik, who cautioned her against leaping too far ahead of the field. Malik was one of Li's heroes, but warned his protégé, "I think you've taken this idea way too far... The trick is to grow with your field. Not to leap so far ahead of it."[\[46\]](#)[\[47\]](#) Such a pursuit was seen as overkill. Researchers were not ready for this kind of scale.

2.4 Why Training is Hard

An artificial neural network (ANN) is said to be learning when it adjusts its weights and biases based on feedback from training data to minimize prediction error. This learning process is guided by a loss function, which measures the difference between the model's predictions and the correct answers. The model adjusts its parameters step by step through gradient descent to minimize this loss and improve its performance.

A critical factor in this process is how the network's weights are initialized. Weight initialization sets the starting values of parameters before training. Random weights are commonly used to ensure that each computational block in the network, known as neurons, learns something unique. If all weights are initialized to the same value, such as zero or any constant, every neuron in a layer receives the same gradient during backpropagation. Consequently, they learn the same features, which clearly restricts the network's ability to model complex patterns. In contrast, random weights encourage the network to learn more complex patterns.

However, if weights start too large, neuron outputs can grow excessively after applying an activation function, causing gradients to explode and neurons to saturate. This instability prevents the model from converging.[\[48\]](#) Gradients can vanish if weights start too small, causing slow or stalled learning.[\[49\]](#) These problems become particularly pronounced as networks grow deeper.

To understand why, consider how gradients are computed during backpropagation. They are the product of multiple partial derivatives as you move backward through the layers. If these derivatives are smaller than one, then the product shrinks with depth, leading to vanishing gradients. The result is that weights in earlier layers receive updates so small they effectively stop learning. Conversely, if the derivatives are greater than one, the gradient can explode, causing erratic weight updates that prevent convergence. Researchers at the time believed that multi-layer networks were fundamentally difficult to train, not because of limitations in data and compute, but because the optimization landscapes were riddled with poor local minima or saddle points, making optimization unstable.

Simply put, before AlexNet, training networks with any depth was considered impractical.

These challenges were amplified by the use of early activation functions like sigmoid and tanh. As gradients propagate backward through multiple layers, they diminish, becoming too weak to update the initial layers of the network. This vanishing gradient problem led many to conclude that depth itself was the problem. Deeper networks didn't just require more time to train and generate higher electricity bills; they performed worse. This phenomenon challenged the very notion that adding more layers would improve performance. It made optimization more difficult, reinforcing the belief that multi-layer networks were inherently unstable and required special tricks.

To escape this dilemma, the dominant approach to training artificial neural networks was to use layer-wise unsupervised pretraining followed by supervised fine-tuning. Instead of training the entire network at once, each layer was trained individually. The output of one layer served as the input to the next, building up the network incrementally. Pretraining brought the network weights much closer to a good solution, allowing backpropagation to succeed where it previously failed. Once all layers had been pretrained in this way, they were "unrolled" into a single architecture.[\[50\]](#) This initialization helped the model avoid poor local minima and improved convergence during training. After unrolling, the entire network could be fine-tuned end-to-end using backpropagation.

This strategy was introduced by Geoffrey Hinton in 2006 and expanded by Yoshua Bengio in 2007, who demonstrated that unsupervised pretraining could effectively unlock deep architectures that were otherwise untrainable using gradient-based methods.[\[51\]](#)[\[52\]](#) These results were striking but also gave rise to a new orthodoxy. Researchers began to believe that deep networks would inevitably fail without pretraining. Against this backdrop of skepticism and stalled progress, James Martens's 2010 paper, *Deep Learning via Hessian-free Optimization*, stood out. It significantly influenced Ilya, who has frequently cited it as one of the key inspirations behind AlexNet.[\[53\]](#) Martens's work directly challenged the prevailing belief that unsupervised pretraining was necessary for training multi-layer networks. Instead, he showed that models could be trained from scratch with random initialization.

Martens reframed the problem. The difficulty wasn't due to abundant local minima, as widely assumed, but stemmed from limitations in first-order methods like gradient descent. He demonstrated that deep networks often contain regions of "pathological curvature." These regions (Figure 2.2) are long, narrow valleys that cause optimization algorithms to stall. Gradient-based methods lack curvature awareness, either oscillating uselessly or making tiny, incremental steps, slowing training dramatically. Martens proposed a second-order approach leveraging curvature to traverse these valleys efficiently. His insights revealed that the actual barrier wasn't the loss landscape itself but rather the inadequate tools to navigate its geometry.[\[54\]](#)

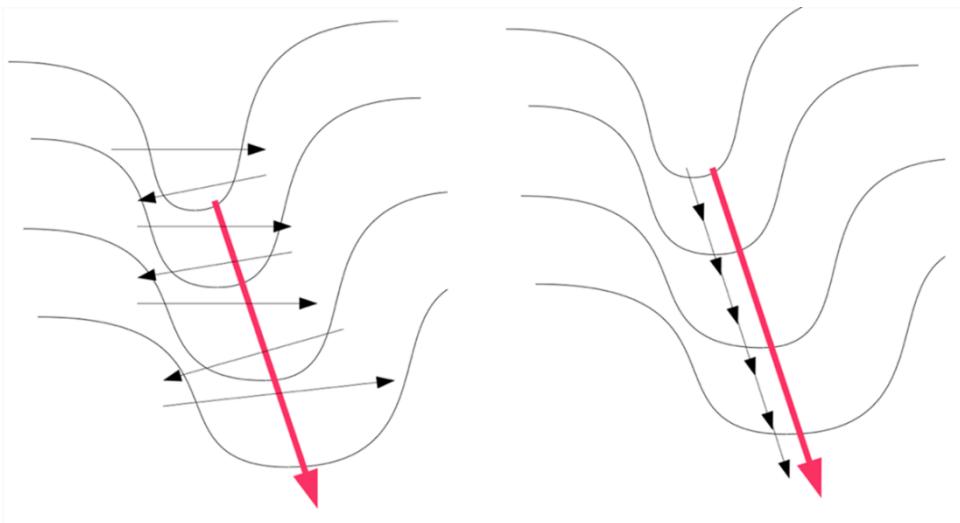


Figure 2.2 The figure illustrates the challenge of optimizing functions shaped as long, narrow valleys. The contour lines represent a valley, with the optimal direction of progress indicated by an arrow running along the valley's base. Smaller arrows represent steps taken by gradient descent: the left diagram shows large steps oscillating inefficiently across the valley, while the right diagram demonstrates small steps making slow progress along the base. The “pathological” nature does not arise from high or low curvature, but from the combination of high curvature across the valley and low curvature along it. *Martens (2010), Fig. 1. Used with explicit permission granted by the author (James Martens).*

Martens introduced a second-order optimization method and applied it successfully to deep autoencoders, models that, until then, had only been trainable with unsupervised layer-wise pretraining. The results were striking. Starting from random initialization, Martens’ method not only converged but also outperformed the same pretrained architecture established by Hinton & Salakhutdinov (2006).^[55] As Martens noted in the paper’s abstract, “Without using pre-training, we obtain results superior to those reported by Hinton & Salakhutdinov (2006) on the same tasks they considered.”

More important than the results was the principle behind them: the success wasn’t due to any specific model architecture or clever initialization scheme. It was due to a more capable optimizer that could navigate the complex loss surfaces. Where traditional methods like gradient descent would stall, Martens’s approach could better understand the shape of the loss landscape and follow more direct paths toward lower error. Martens had proven that unsupervised pretraining wasn’t necessary. The real obstacle wasn’t depth but the lack of tools to deal with curvature.

Sutskever recognized this immediately. While many researchers still clung to pretraining or handcrafted features, he saw that this research had cracked something fundamental. Martens acknowledges Sutskever in the 2010 paper, and the two even collaborated in 2011 to apply second-order optimization to training recurrent neural networks (RNNs), another notoriously difficult architecture.[\[56\]](#)[\[57\]](#) Though Martens's method was computationally heavy and better suited for smaller problems, Sutskever understood the broader implication: if a second-order method could train a deep network from scratch, then a first-order method with the right refinements might be able to do the same at scale.

That became the bet behind AlexNet. Improved initialization, better activation functions, and sufficient computational power allowed gradient descent to overcome the obstacles that once made deep learning seem impossible. In later interviews, Sutskever claimed he *knew* AlexNet would work. It might sound like hindsight dressed up as confidence to some, but it wasn't. Martens had already shown that deep networks could converge. For Sutskever, the confidence wasn't theoretical; it was empirical. He had already seen it happen.[\[58\]](#)

2.5 AlexNet

AlexNet redefined the ImageNet competition with a top-5 error of 15.3%. AlexNet didn't merely surpass the runner-up with 26.2%; it obliterated it. These results emerged from carefully engineered architectural innovations detailed in the seminal paper *ImageNet Classification with Deep Convolutional Neural Networks*, commonly known as the "AlexNet paper" and often abbreviated as KSH after authors Krizhevsky, Sutskever, and Hinton.[\[59\]](#) AlexNet significantly increased the depth and width of the standard network configuration, leveraged Graphic Processing Units (GPUs) to manage computational demands, and trained the model on the large ImageNet dataset. Crucially, KSH introduced several innovations, including ReLU activations, dropout regularization, and extensive data augmentation.

This dramatic leap challenges earlier claims, such as Efros's 'stone soup' analogy and *The Unreasonable Effectiveness of Data*, which promote data as a universal equalizer. If that were true, one would expect a gradual performance improvement across the board with more data. However, prior to AlexNet, ImageNet participants relied exclusively on handcrafted features and shallow methods like SVMs yet experienced only marginal gains between 2010 and 2011. In contrast, AlexNet's success wasn't incremental. It halved the error rates and ushered in rapid improvements in subsequent years. This empirical evidence demonstrates convincingly that while large-scale data is indeed necessary, it alone is insufficient: artificial neural networks uniquely and disproportionately benefit from scale, exposing the limitations inherent in the stone soup analogy and the unreasonable effectiveness of data.

Structurally, AlexNet consists of 60 million parameters and about 650,000 so-called “neurons.” As previously discussed, parameters are the weights and biases that the model tunes during training. Neurons, meanwhile, refer to the computational blocks of the model that take some input multiplied by weights, potentially add a bias term, and pass the result through an activation function that rescales the raw output to a specific range (0 to 1 or -1 to 1). A single neuron isn’t very useful. However, connecting many neurons across multiple layers allows the network to learn complex patterns from data, such as recognizing a cat or translating languages. Typically, researchers emphasize parameter counts because they directly influence how challenging the model is to train, how expensive it is to run, and how easily it might overfit. Neuron counts, by contrast, are more abstract and ambiguous, particularly in convolutional layers where parameters are shared.

Yet, despite this ambiguity, KSH explicitly highlighted its neuron count. Why highlight this seemingly abstract measure? In 2012, artificial neural network researchers often emphasized analogies to biological systems to provide intuition and motivation for their architectural choices. AlexNet firmly positioned itself within this biologically inspired tradition, explicitly referencing neuron counts and promoting local response normalization, a concept the authors claim was inspired by the brain’s lateral inhibition, despite its limited practical impact on AlexNet’s performance.

Such parallels to biology provided essential legitimacy in an era before dramatic improvements on standardized benchmarks rendered them largely unnecessary. Today, neural networks are generally described in pragmatic, engineering-focused terms. Still, AlexNet’s explicit neuron count reflects a time when biological intuition was crucial to establishing credibility and inspiring confidence in novel approaches.

2.5.1 Network Architecture

At the core of AlexNet’s success are its first five convolutional layers, which enable the automatic hierarchical feature extraction. Unlike traditional handcrafted methods, the convolutional layers progressively capture increasingly complex patterns starting from simple patterns such as edges and textures in the earliest layers, evolving into intricate shapes, corners, and recognizable object parts deeper in the network.

Each convolutional layer employs multiple small filters that slide across the input image, performing dot products between filter weights and input pixels. This produces feature maps highlighting precisely where specific visual patterns occur. Crucially, these filters share weights across the entire input, reducing parameters dramatically and ensuring translation invariance, which is the ability to detect patterns wherever they appear. For instance, AlexNet’s first convolutional layer applies 96 distinct filters, generating 96 unique feature maps that feed into deeper layers for further abstraction and analysis.

AlexNet's convolutional layers weren't merely deep. They were strategically wide, too. Each successive convolutional layer dynamically increased its filter count, from 96 filters in the first layer to 256, 384, 384, and finally 256, significantly expanding the network's capacity. At the same time, filter sizes decreased from larger 11x11 filters in initial layers to 5x5 filters, eventually using 3x3 filters deeper into the network. This progression allowed AlexNet to focus initially on coarse features and then refine these into increasingly intricate, fine-grained patterns. Max pooling complemented this strategy by systematically reducing feature map dimensions, selecting only the strongest activations within small spatial regions, typically 3x3 windows. This selective reduction decreased computational complexity and enhanced the network's robustness to minor positional variations or input distortions. By narrowing filter size and increasing the quantity of feature maps in deeper layers, AlexNet expanded each neuron's receptive field or the region of the input image influencing that neuron's activation. This allowed the network to build layered representations, capturing increasingly complex visual abstractions across broader image regions.

After the convolutional layers have extracted features, AlexNet transitions into three fully connected (FC) layers. This is the network's critical final step toward image classification. Unlike convolutional layers, FC layers connect every neuron to all neurons in the preceding layer. This dense connectivity flattens the feature maps into a single vector, intentionally discarding spatial structure to integrate high-level features into a unified, comprehensive representation. By doing so, the FC layers can synthesize information across the entire image, identifying global patterns essential for accurate classification. This shift from spatially aware feature extraction to spatially agnostic global interpretation equips AlexNet to map complex visual features effectively into precise, categorical predictions.

AlexNet's first two FC layers constitute most of the model's roughly 60 million parameters. This scale dramatically increased the model's representational capacity but risked overfitting. To address this, AlexNet employed a novel technique known as dropout, randomly deactivating 50% of neurons during training to prevent excessive co-adaptation. The final FC layer outputs a 1,000-dimensional vector corresponding directly to the 1,000 categories in the ImageNet dataset. A softmax function transforms these raw scores into predicted probabilities, with the highest probability determining the network's final classification. In sum, convolutional layers extract visual patterns, and fully connected layers interpret these patterns into categorical predictions. Together, they form a computational pipeline that, while devoid of genuine phenomenological experience or visual understanding, enables AlexNet to translate raw pixels into object classifications with groundbreaking accuracy.

2.5.2 Training Innovations

A critical innovation in AlexNet, influenced by insights from Ilya Sutskever, was the adoption of the Rectified Linear Unit (ReLU) activation after each convolutional layer.[\[60\]](#) Unlike traditional activation functions such as sigmoid or tanh, which compress outputs into limited ranges and often cause neuron saturation and vanishing gradients, ReLU maintains strong gradient flow throughout training. Figure 2.3 illustrates that ReLU achieves this by outputting zero for negative inputs and preserving positive inputs unchanged, thus acting like an identity function. This seemingly simple modification enabled AlexNet to scale effectively to previously unseen depths.

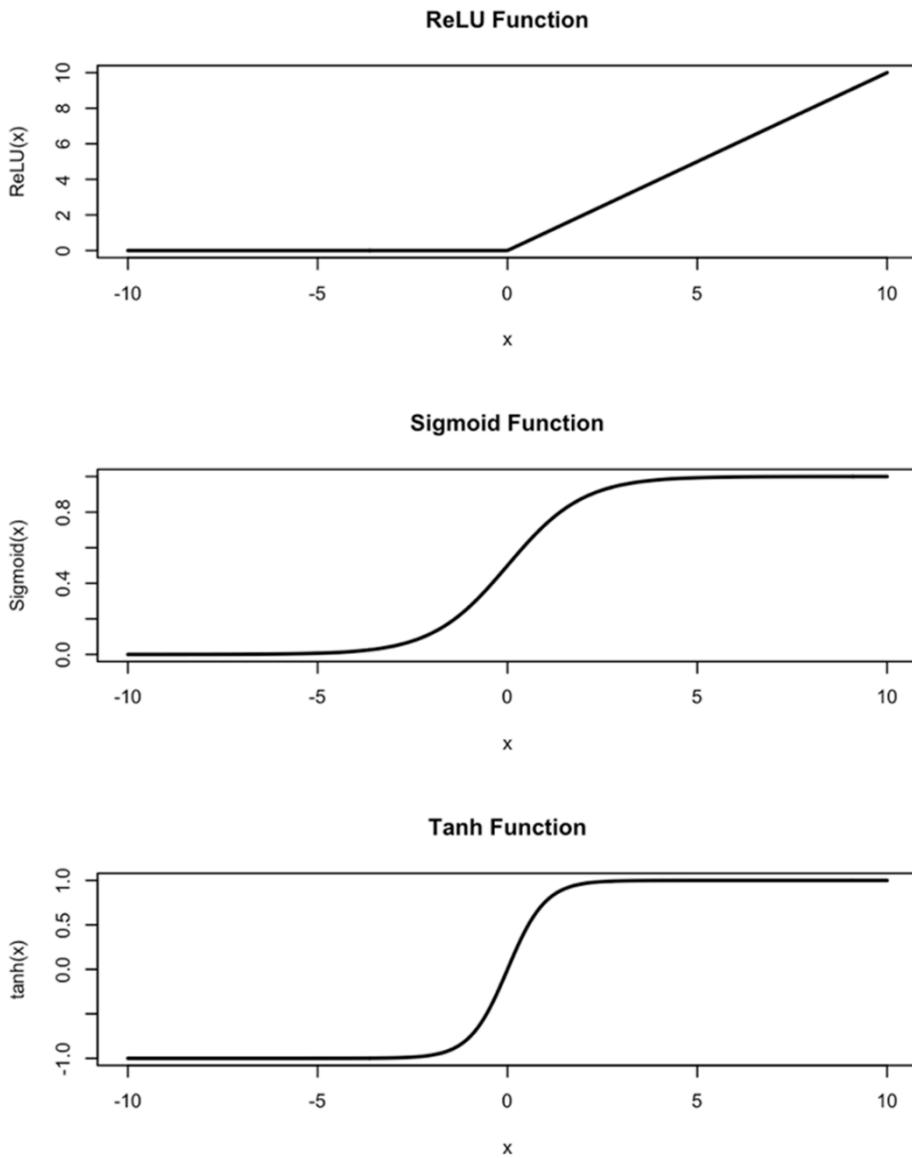


Figure 2.3 Plot of ReLU (top), sigmoid (middle), and tanh (bottom) activation functions generated in R by the author. Activation functions map input values (on the x-axis) to output values (on the y-axis). For example, ReLU produces zero for negative inputs; for positive inputs, the output equals the input ($x < 0$ and linear for $x \leq 0$). Sigmoid smoothly transitions inputs into values between 0 and 1, shaped like an S-curve, and tanh similarly provides a smooth S-shaped curve but maps inputs into values ranging between -1 and 1, centering around zero.

In a controlled experiment, the KSH demonstrated that switching from traditional tanh activations to ReLU produced a remarkable sixfold acceleration in training convergence. The authors recognized how critical this efficiency gain was, noting explicitly that “we would not have been able to experiment with such large [artificial] neural networks for this work if we had used traditional saturating neuron (i.e., sigmoid or tanh activation functions) models.”[\[61\]](#) This improvement was possible because ReLU activations flattened the optimization landscape by avoiding saturation and alleviating what Martens described as “pathological curvature.” Additionally, ReLU also induced sparse activations, meaning many neurons output zeros, which acted as a form of regularization by reducing inter-neuronal co-adaptation, contributing to the overall impact of ReLU on training.

AlexNet’s success firmly established ReLU as the default activation for deep networks. Later networks such as VGGNet, GoogLeNet, and ResNet all employed ReLU or its variants, confirming that non-saturating activation was critical for scaling to greater depth. AlexNet also initialized biases for some convolutional layers to 1 to ensure ReLU outputs started active. Combining ReLUs and careful weight initialization helped avoid dead neurons and facilitated stable training.

Fully connected layers contained most of AlexNet’s parameters, making them especially prone to overfitting. To address this vulnerability, the authors employed dropout, which randomly disables a fraction of neurons during each training iteration. In AlexNet’s case, 50% of neurons in the first two fully connected layers (FC6 and FC7) were dropped on each training iteration. Geoffrey Hinton, the godfather of strained analogies, famously illustrated dropout’s intuition using the “co-conspirator” analogy. If you want to find out who committed a crime, you interrogate suspects separately, not together. Similarly, dropout forced neurons to independently learn useful features rather than relying on tightly coupled groups.

KSH noted that the network “exhibits substantial overfitting” without dropout. With 60 million parameters and just 1.2 million training images, the network could easily memorize rather than generalize. Dropout provided a form of regularization that dramatically improved generalization. An interesting side effect was that it slowed training: “Dropout roughly doubles the number of iterations required to converge,” the paper reports. This makes sense since, with dropout, each training update is a bit less efficient, and each neuron sees only half the updates it usually would. But this is a small price for the gain in test performance. Following AlexNet’s success, dropout rapidly became a standard practice in CNNs whenever overfitting was a concern, showcasing the impact of a simple yet powerful idea for improving network generalization.

As previously noted, AlexNet also introduced Local Response Normalization (LRN), an experimental technique inspired by biological lateral inhibition. LRN normalizes each neuron’s activation relative to neighboring neurons at the same spatial location across adjacent feature maps, encouraging competition among neurons. This helped suppress uniformly strong activations and enhanced distinctive, selective feature responses. In practice, the authors observed only modest improvements in generalization but still included LRN after the first two convolutional layers. However, LRN soon fell out of favor and was replaced by more effective normalization methods, notably batch normalization in 2015.[\[62\]](#) While modern networks no longer use LRN, its inclusion in AlexNet exemplifies the spirit of experimental innovation characteristic of its design, even if, in hindsight, AlexNet’s results were not critically dependent on this particular technique.

2.5.3 Data Augmentation

KSH used aggressive data augmentation to expand ImageNet's effective size and diversity. Two primary techniques were employed:

1. Geometric Transformations: Random crops and horizontal reflections were employed to teach the network spatial invariance, enabling it to recognize objects even when they appeared in various positions or orientations.
2. Photometric Distortions: PCA-based color jittering introduced subtle variations in brightness, contrast, and color balance by perturbing the principal components of the RGB pixel values. This technique simulated the effects of lighting changes and enhanced robustness to illumination variance.

Together, these augmentation strategies improved AlexNet's generalization by simulating realistic variability and noise within the training data itself. The combination of data augmentation and dropout effectively mitigated overfitting, even though the model was highly over-parameterized, with 60 million parameters trained on just 1.2 million images. This allowed the network to leverage its capacity, precisely fitting the training data while still generalizing successfully to unseen test images from the ImageNet dataset.

2.5.4 Efficient and Scalable Training

GPUs were initially designed for fast-moving video game graphics, but the calculations they excel at closely resemble those needed for training artificial neural networks. At the time, NVIDIA searched for new applications for its niche hardware. In an interview with *MIT Technology Review*, NVIDIA CEO Jensen Huang says, "When you invent a new technology, you have to be receptive to crazy ideas." He continues, "My state of mind was always looking for something quirky, and the idea that [artificial] neural networks would transform computer science—that was an outrageously quirky idea."[\[63\]](#)

Training an 8-layer network with 60 million parameters on 1.2 million images wasn't just computationally demanding; it was conceptually difficult. The NVIDIA GTX 580 Graphic Processing Units (GPUs) available in 2012 had only 3GB of memory, which was insufficient to store the entire model. KSH strategically split the network across the two GPUs to overcome this constraint. Each GPU processed half the filters in every layer and half the feature maps for each training batch. Inspired by earlier model parallelization research, this two-column architecture minimized overhead by synchronizing GPU communication at carefully chosen layers: after the second convolutional layer to enable cross-GPU learning once early processing was complete, and after the fifth, to integrate features fully before transitioning to the network's fully connected layers.[\[64\]](#)[\[65\]](#)[\[66\]](#)[\[67\]](#)[\[68\]](#)

Utilizing two GPUs in parallel enabled the team to scale the network, allowing for twice as many filters per layer as a smaller one-GPU variant. This enhanced top-1 accuracy by 1.7% and top-5 accuracy by 1.2%. The two-GPU setup also enabled greater depth, which is crucial to its performance. KSH found that removing any convolutional layer and retraining decreased accuracy, even though each convolutional layer accounted for less than 1% of the model's parameters.

These observations demonstrated that AlexNet's strength derived partially from its increased size and depth, facilitating multi-stage feature extraction. Modern convolutional neural networks, now routinely trained on GPU clusters, directly extend the trajectory AlexNet established. At the time, the team explicitly recognized GPU memory and computational time as their primary constraints, stating that the network's scale was "limited mainly by the amount of memory available on current GPUs and by the amount of training time we were willing to tolerate." They anticipated further improvements, noting their "results can be improved simply by waiting for faster GPUs and bigger datasets to become available." Despite the comparatively modest scale of their original work, subsequent advances have fully validated this prediction, underscoring AlexNet's influence toward deeper and more powerful networks.

Training neural networks effectively requires carefully balancing several factors to ensure stable learning and optimal performance. AlexNet was trained with mini-batch stochastic gradient descent (SGD) augmented by momentum 0.9. In first-order methods like stochastic gradient descent (SGD), the model relies solely on the gradient, representing the slope of the loss function, to determine the adjustments to the model's parameters (i.e., weights). Specifically, the gradient indicates the direction and rate of steepest increase in the loss; thus, first-order methods move in the opposite direction (i.e., downhill), tweaking weights to reduce the loss. In other words, first-order optimization uses only information about how the function changes locally, at the current point, without needing additional details such as curvature (i.e., second-order derivatives). This simplicity makes first-order methods computationally efficient and popular for training large networks.

Imagine a lone hiker on a foggy mountain. Since the hiker cannot see the whole terrain, they scan a small patch of ground around them, representing a random mini-batch of images to gauge the local downhill direction, and then move a short distance opposite that slope. Because the scan is only a sample, the estimate is noisy. Still, momentum acts like inertia, averaging recent steps so the hiker glides through shallow plateaus instead of taking slightly uphill steps or jitters sideways.

AlexNet's training used 128-image batches, an initial learning rate of 0.01, and a manual schedule that dropped the rate by a factor of 10 whenever validation accuracy flattened. This lets the hiker take bold steps early on and finer, deliberate steps near the valley floor. Although SGD never guarantees the absolute lowest point. In rugged landscapes, SGD can zig-zag, get stuck on plateaus or saddle points, and never guarantee reaching the global minimum. However, the combination of mini-batch sampling, momentum, and a decaying learning rate reliably carried AlexNet into a deep enough basin to achieve its breakthrough ImageNet performance.

Careful weight initialization also contributed to AlexNet's training stability. Weights were initialized from a zero-mean Gaussian distribution with a small standard deviation (0.01), ensuring neuron outputs began training within a stable, balanced range. Furthermore, biases for specific layers were set intentionally to 1 instead of the typical 0, thereby activating ReLU neurons at the training outset. This strategy mitigated the "dying ReLU" problem, where inactive units get stuck outputting zero and never recover.[\[69\]](#) Although minor, this careful initialization was crucial for maintaining healthy training dynamics.

Despite AlexNet's eventual success using gradient descent, many researchers initially doubted that this approach could train multi-layer artificial neural networks. A prevalent misconception at the time was that gradient-based optimization methods would inevitably become trapped in local minima, rendering training ineffective. Researchers visualized the loss landscape as a low-dimensional terrain filled with poor local minima. However, they overlooked that loss surfaces in high-dimensional spaces behave differently; moreover, by averaging gradients over mini-batches, as AlexNet explicitly did, the training process further smooths irregularities in the loss landscape, facilitating more stable optimization. Recognizing this subtle yet profound truth was essential in enabling the deep learning revolution that AlexNet pioneered.

2.5.5 The Fallout

James Martens showed that multi-layer networks could be trained from scratch and identified why previous attempts had failed. This gave Ilya the confidence to move forward. AlexNet's architecture included a series of optimization-aware choices: ReLU activations to prevent vanishing gradients, proper weight initialization to stabilize early training, momentum to smooth updates, dropout to reduce overfitting, and data augmentation to increase the effective size of the training set.[\[70\]](#) While these techniques weren't individually new, KSH combined them to enable end-to-end supervised learning at scale. AlexNet was also distributed across two GPUs to make training feasible. This humble pair of 3GB GPUs illustrates a profound truth. Everything has humble beginnings, and the "deep learning revolution" is no exception.

Fei-Fei Li later called AlexNet "symbolic," highlighting three fundamental elements converging for the first time: large-scale data (ImageNet), powerful GPUs, and improved optimization-aware algorithms.[\[71\]](#) It was the recipe needed to revive artificial neural networks. However, Li's reflection omits the human aspect. A small group of researchers maintained conviction even as broader skepticism dominated the field. This conviction was essential to AlexNet's eventual breakthrough.

AlexNet sparked an immediate and dramatic shift. By 2013, the ImageNet competition (ILSVRC) attracted twenty-four teams, more than the previous three years combined. Almost all entries used convolutional neural networks, marking a remarkable turnaround for an approach considered impractical a year earlier. The zeitgeist had shifted entirely: not only had artificial neural networks shed their stigma, but they were now driving the state of the art. In 2011, you'd be rejected for using ANNs; by 2013, you might be rejected for not using them. The "vogue" had flipped entirely.

AlexNet's success inspired a wave of deeper and more effective architectures. In 2013, a team from NYU won ILSVRC 2013 by introducing the "AlexNet++" (ZFNet), achieving a top-5 error of 11.2%.[\[72\]](#) In 2014, accuracy against ImageNet surpassed 90%. That year, GoogLeNet won ILSVRC with an astonishing 6.7% top-5 error rate, which represents roughly a 10-fold reduction in error in just two years compared to pre-AlexNet results. The momentum continued: by 2017, 76% (29 out of 38) of teams competing in ILSVRC achieved greater than 95% top-5 accuracy.[\[73\]](#),[\[74\]](#),[\[75\]](#) Once deemed too ambitious by Jitendra Malik, the ImageNet benchmark had become saturated and effectively solved in just five years, prompting organizers to discontinue the competition.

After AlexNet's success, Malik embraced deep learning. In 2014, his group at UC Berkeley developed R-CNN, an approach that applied CNNs to achieve state-of-the-art object detection.^[76] This rapid follow-up showed the AlexNet moment was no fluke. AlexNet could be replicated and extended by others. In 2017, Malik called AlexNet "the most impactful paper in machine learning and computer vision in the last five years," calling it "the paper that led the field of computer vision to embrace deep learning."^[77] In 2020, Malik admitted to being "positively surprised" by the effectiveness of deep learning, confessing that he "certainly would not have believed that in 2010."^[78]

Beyond academia, AlexNet catalyzed a strategic transformation in the tech industry. Google moved first by acquiring DNNresearch in early 2013, establishing Google Brain and acquiring DeepMind. Facebook responded by launching Facebook AI Research in late 2013, appointing CNN pioneer Yann LeCun as its director. Baidu recruited Andrew Ng to lead a new AI lab on speech and image tasks. Clarifai, whose founders won ILSVRC 2013, emerged to provide commercial CNNs. AlexNet did not merely win a benchmark; it redefined the priorities of the tech sector.

AlexNet also dramatically reshaped public narratives around AI. MIT Technology Review declared "Deep Learning" among its top breakthroughs of 2013, declaring AI was "finally getting smart."^[79] That optimism surged by the year's end when *The New York Times* featured a story titled "Brainlike Computers, Learning From Experience" on its front page, signaling how profoundly deep learning had captivated the cultural imagination.^[80] Even critical coverage, such as *The Guardian's* "Don't believe the science hype – we haven't created true AI yet," indirectly validated the new paradigm's dominance.^[81]

AlexNet woke a dormant field from decades of malaise. Its success demonstrated for the first time that artificial neural networks could significantly outperform traditional methods, hinting at the potential of scale. This early indication gradually evolved into the modern era, where scaling in terms of data, model size, and computation has become central to advances in artificial intelligence. The few who still expressed skepticism did so based on the so-called "black box" criticism, where models lack understanding or excessive hype, but not on actual performance.

- [1] http://vision.stanford.edu/cs598_spring07/papers/Lecun98.pdf
- [2] Hopfield JJ. Neural networks and physical systems with emergent collective computational abilities. Proc Natl Acad Sci U S A. 1982 Apr;79(8):2554-8. doi: 10.1073/pnas.79.8.2554. PMID: 6953413; PMCID: PMC346238.
- [3] <https://people.idsia.ch/~juergen/DanNet-triggers-deep-CNN-revolution-2011.html>
- [4] LeCun, Y., Bengio, Y. & Hinton, G. Deep learning. *Nature* 521, 436–444 (2015). <https://doi.org/10.1038/nature14539>
- [5] <https://www.nature.com/articles/323533a0>
- [6] <https://stacker.news/items/762990>
- [7] (pg. 4) <https://web.cs.ucla.edu/~kaoru/FordMR2018Architects-of-Intelligence.pdf#:~:text=significant%20enthusiasm%20among%20researchers,1960s%2C%20however—at%20least%20in%20part>

- [8] Cortes, C., Vapnik, V. Support-vector networks. *Mach Learn* 20, 273–297 (1995). <https://doi.org/10.1007/BF00994018>
- [9] Yoav Freund, Robert E Schapire, A Decision-Theoretic Generalization of On-Line Learning and an Application to Boosting, Journal of Computer and System Sciences, Volume 55, Issue 1, 1997, Pages 119-139, ISSN 0022-0000, <https://doi.org/10.1006/jcss.1997.1504>.
- [10] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001, Kauai, HI, USA, 2001, pp. I-I, doi: 10.1109/CVPR.2001.990517.
- [11] <https://thegradient.pub/the-limitations-of-visual-deep-learning-and-how-we-might-fix-them#:~:text=We%20are%20witnessing%20the%20third,GPUs%29%20became%20available>
- [12] <https://www.cs.toronto.edu/~hinton/absps/NatureDeepReview.pdf#:~:text=mainstream%20computer,different%20classes%2C%20they%20achieved%20spectacular>
- [13] <https://www.cs.toronto.edu/~hinton/absps/ncfast.pdf>
- [14] Bengio, Y., Lamblin, P., Popovici, D., and Larochelle, H. Greedy layer-wise training of deep networks. In NIPS, 2007.
- [15] (pg.4) <https://web.cs.ucla.edu/~kaoru/FordMR2018Architects-of-Intelligence.pdf#:~:text=dismissed%20as%20researchers%20embraced%20other,advance%20the%20technology%20of%20neural>
- [16] <https://www.computervisionblog.com/2015/01/from-feature-descriptors-to-deep.html#:~:text=Visual%20Words%20were%20introduced%20by,heavily%20utilized%20in%20vision%20research>
- [17] <https://www.cs.ubc.ca/~lowe/papers/ijcv04.pdf>
- [18] Bay, H., Tuytelaars, T., Van Gool, L. (2006). SURF: Speeded Up Robust Features. In: Leonardis, A., Bischof, H., Pinz, A. (eds) Computer Vision – ECCV 2006. ECCV 2006. Lecture Notes in Computer Science, vol 3951. Springer, Berlin, Heidelberg. https://doi.org/10.1007/11744023_32
- [19] Csurka, G., Dance, C., Fan, L., Williamowski, J., & Bray, C. (2004). Visual categorization with bags of keypoints. In Proceedings of ECCV2004 Workshop on Statistical Learning in Computer Vision (pp. 59–74). <https://www.cs.cmu.edu/~efros/courses/LBMV07/Papers/csurka-eccv-04.pdf>
- [20] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05), San Diego, CA, USA, 2005, pp. 886-893 vol. 1, doi: 10.1109/CVPR.2005.177.
- [21] <https://pmc.ncbi.nlm.nih.gov/articles/PMC5876362/#:~:text=from%20AlexNet13%20pre,features%2C%20which%20can%20transfer%20well>

- [22] <https://arxiv.org/pdf/1411.4304>
- [23] <https://cs.brown.edu/people/pfelzens/papers/lsvm-pami.pdf>
- [24] <https://www.computervisionblog.com/2015/01/from-feature-descriptors-to-deep.html#:~:text=DPM%20was%20the%20reigning%20champ,apart%20the%20DPM%20codebase%20several>
- [25] <https://link.springer.com/article/10.1007/s11263-014-0733-5>
- [26] Jia, Yangqing & Huang, Chang & Darrell, T.. (2012). Beyond spatial pyramids: Receptive field learning for pooled image features. Proceedings / CVPR, IEEE Computer Society Conference on Computer Vision and Pattern Recognition. IEEE Computer Society Conference on Computer Vision and Pattern Recognition. 3370-3377. 10.1109/CVPR.2012.6248076.
- [27] https://www.ri.cmu.edu/pub_files/2012/8/divvalaThesis.pdf
- [28] <https://youtu.be/ElyFDUab30A?si=j48mzXqevdfY5oJV&t=1640>
- [29] Vondrick, Carl, Aditya Khosla, Tomasz Malisiewicz, and Antonio Torralba. "HOGgles: Visualizing Object Detection Features." 2013 IEEE International Conference on Computer Vision, 1-8 Dec. 2013, Sydney, NSW. (December 2013). p.1-8.
- [30] <https://scilogs.spektrum.de/hlf/data-is-unreasonably-effective-and-there-is-plenty-of-it-to-explore/#:~:text=Efros%20shared%20examples%20from%20his,image%20or%20guessing%20where%20a>
- [31] A. Torralba and A. A. Efros, "Unbiased look at dataset bias," CVPR 2011, Colorado Springs, CO, USA, 2011, pp. 1521-1528, doi: 10.1109/CVPR.2011.5995347. https://people.csail.mit.edu/torralba/publications/datasets_cvpr11.pdf
- [32] <https://thegradient.pub/the-limitations-of-visual-deep-learning-and-how-we-might-fix-them/#:~:text=We%20are%20witnessing%20the%20third,GPUs%29%20became%20available>
- [33] Li, Dr. Fei-Fei. The Worlds I See: Curiosity, Exploration, and Discovery at the Dawn of AI (p. 194). Kindle Edition.
- [34] <https://www.youtube.com/watch?v=dV6sgUiGv-U>
- [35] <https://scilogs.spektrum.de/hlf/data-is-unreasonably-effective-and-there-is-plenty-of-it-to-explore/#:~:text=algorithms%20-%20transformers%2C%20deep%20learning%2C,the%20data%20that%20powers%20them>
- [36] A. Halevy, P. Norvig and F. Pereira, "The Unreasonable Effectiveness of Data," in IEEE Intelligent Systems, vol. 24, no. 2, pp. 8-12, March-April 2009, doi: 10.1109/MIS.2009.36.
- [37] https://www.facebook.com/story.php?story_fbid=pfbid0b6rC7KJFHvxLjjbPtHyz53Ffjy9YWRj3uQc6taVpv9CdqDUZ2z2ywGDhdgZKx5HI&id=1533402400

- [38] <https://www.newsroom.hlf-foundation.org/blog/article/data-is-unreasonably-effective-and-there-is-plenty-of-it-to-explore/>
- [39] <https://simons.berkeley.edu/news/stone-soup-ai>
- [40] <https://en.wikipedia.org/wiki/AlexNet#:~:text=In%202011%2C%20Geoffrey%20Hinton%20.19>
- [41] Li, Fei Fei (2023). The worlds I see: curiosity, exploration, and discovery at the dawn of AI (First ed.). New York: Moment of Lift Books ; Flatiron Books.
- [42] Li, Dr. Fei-Fei. The Worlds I See: Curiosity, Exploration, and Discovery at the Dawn of AI (p. 201). (Function). Kindle Edition.
- [43] <https://slow-thoughts.com/seeing-the-light/>
- [44] Biederman, Irving (1987). "Recognition-by-components: A theory of human image understanding". *Psychological Review*. 94 (2): 115–117. doi:10.1037/0033-295X.94.2.115. ISSN 0033-295X. PMID 3575582.
- [45] <https://link.springer.com/article/10.1007/s11263-014-0733-5>
- [46] This narrative is explored in detail here: <https://slow-thoughts.com/seeing-the-light/>
- [47] <https://stacker.news/items/762990>
- [48] <https://arxiv.org/abs/1712.05577>
- [49] <https://www.cs.toronto.edu/~hinton/absps/science.pdf>
- [50] Erhan, D., Bengio, Y., Courville, A., Manzagol, P., Vincent, P., and Bengio, S. Why does unsupervised pre-training help deep learning? *Journal of Machine Learning Research*, 2010.
- [51] Hinton, G. E. and Salakhutdinov, R. R. Reducing the dimensionality of data with neural networks. *Science*, July 2006.
- [52] Bengio, Y., Lamblin, P., Popovici, D., and Larochelle, H. Greedy layer-wise training of deep networks. In *NIPS*, 2007.
- [53] e.g., Lex Fridman Podcast: <https://youtu.be/13CZPWmke6A?si=sSjdJMX670GeINFZ&t=209>
- [54] As Martens put it: "The difficulty associated with learning deep autoencoders is better explained by regions of pathological curvature, which to first-order optimization methods resemble bad local minima."
- [55] Martens, J. (2010). *Deep learning via Hessian-free optimization*. Proceedings of the 27th ICML. (https://www.cs.toronto.edu/~jmartens/docs/Deep_HessianFree.pdf)
- [56] https://icml.cc/Conferences/2011/papers/532_icmlpaper.pdf
- [57] https://pdfs.semanticscholar.org/6503/e264e4fdecb22ab0f5d081123814ec242d86.pdf?utm_source=chatgpt.com

- [58] (5:13) <https://youtu.be/13CZPWmke6A?si=pnioUssu8jioetYP>
- [59] https://proceedings.neurips.cc/paper_files/paper/2012/file/c399862d3b9d6b76c8436e924a68c45b-Paper.pdf
- [60] Earlier research by Nair and Hinton (2010) introduced ReLUs for Restricted Boltzmann Machines, and Jarrett et al. (2009) explored alternative activation functions in smaller CNNs.
- [61] AlexNet paper
- [62] <https://arxiv.org/abs/1502.03167>
- [63] <https://www.technologyreview.com/2023/10/26/1082398/exclusive-ilya-sutskever-openai-chief-scientist-on-his-hopes-and-fears-for-the-future-of-ai/>
- [64] D.C. Ciresan, U. Meier, J. Masci, L.M. Gambardella, and J. Schmidhuber. High-performance neural networks for visual object classification. Arxiv preprint arXiv:1102.0183, 2011.
- [65] LeCun, Y., Bengio, Y. & Hinton, G. Deep learning. *Nature* 521, 436–444 (2015). <https://doi.org/10.1038/nature14539>
- [66] https://proceedings.neurips.cc/paper_files/paper/2012/file/c399862d3b9d6b76c8436e924a68c45b-Paper.pdf
- [67] <https://youtu.be/13CZPWmke6A?si=Xpxwch-yiSPMS-Y7&t=142>
- [68] LeCun, Y., Bengio, Y. & Hinton, G. Deep learning. *Nature* 521, 436–444 (2015). <https://doi.org/10.1038/nature14539>
- [69] <https://arxiv.org/abs/1903.06733>
- [70] <http://proceedings.mlr.press/v28/sutskever13.html>
- [71] <https://arstechnica.com/ai/2024/11/how-a-stubborn-computer-scientist-accidentally-launched-the-deep-learning-boom/>
- [72] <https://arxiv.org/abs/1311.2901>
- [73] <https://www.economist.com/special-report/2016/06/23/from-not-working-to-neural-networking>
- [74] He, Kaiming; Zhang, Xiangyu; Ren, Shaoqing; Sun, Jian (2016). "Deep Residual Learning for Image Recognition". *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. pp. 770–778.
- [75] Aron, Jacob (21 September 2015). "Forget the Turing test – there are better ways of judging AI". *New Scientist*. <https://www.newscientist.com/article/dn28206-forget-the-turing-test-there-are-better-ways-of-judging-ai/>
- [76] https://www.cv-foundation.org/openaccess/content_cvpr_2014/papers/Girshick_Rich_Feature_Hierarchies_2014_CVPR_paper.pdf

- [77] <https://cacm.acm.org/research/technical-perspective-what-led-computer-vision-to-deep-learning/#:~:text=It%20is%20my%20opinion%20the,vision%20to%20embrace%20deep%20learning>
- [78] (01:26:57) https://youtu.be/LRYkH-fAVGE?si=B0Bg0BDOuXAjZ_jO&t=5214
- [79] <https://www.technologyreview.com/2013/04/23/178729/the-10-breakthrough-technologies-of-2013/>
- [80] <https://www.nytimes.com/2013/12/29/science/brainlike-computers-learning-from-experience.html>
- [81] https://www.theguardian.com/commentisfree/2014/jun/11/ai-eugene-goostman-artificial-intelligence?utm_source=chatgpt.com

3 Residual Revolution (ResNet)

This chapter covers

- The challenges of training deep neural networks
- Residual connections and how they revolutionized deep learning
- ResNet's breakthrough results on ImageNet and other benchmarks
- How ResNet reshaped scalability and optimization
- Real-world impact and industry adoption of ResNet models
- Extending ResNet with dilated convolutions for dense prediction tasks
- Evolution to ResNet-v2 and training networks exceeding 1,000 layers

Papers

- *Deep Residual Learning for Image Recognition* (2015) He, Zhang, Ren, and Sun
- *Identity Mappings in Deep Residual Networks* (2016) He, Zhang, Ren, and Sun
- *Multi-Scale Context Aggregation by Dilated Convolutions* (2016) Yu and Koltun

In 2015, “deep learning” was a misnomer. Despite the success of AlexNet, network depth remained shallow. The heart of the problem were two knotted and relentless challenges that involved signal loss, but in complementary ways that compounded each other, making them doubly binding. Just when despair seemed justified, a deceptively simple idea emerged.

Residual connections create shortcuts that preserve the integrity of the original signal, protecting it from cumulative distortions and maintaining strong gradient signals throughout backpropagation. ResNet earns its place on *Sutskever's List* because it represents a pivotal shift in design philosophy, where “add a residual connection” has become an indispensable tool in the machine learning toolkit.[\[1\]](#)

The results were astonishing. Like AlexNet, ResNet halved the top-5 error rate on ImageNet from the previous year. At the 2015 ImageNet Large Scale Visual Recognition Challenge, an ensemble of ResNets achieved a mere 3.57% error rate. A single ResNet model realized an extraordinary 4.49%, which was better than all previous ensemble results. ResNet also surpassed the widely touted human-level accuracy threshold. It would be the last time ImageNet saw a step change in performance. Reflecting on this breakthrough, the ResNet team leader Kaiming He confessed, “We didn’t even believe this single idea could be so significant.” This sentiment underscores an important truth in machine learning: the significance of an idea doesn’t correlate with its complexity.

Remarkably, residual connections also delivered practical efficiency. ResNet-152 has 152 layers and is eight times deeper than the nineteen-layer VGGNet, yet it requires fewer computational operations and model parameters. In other words, ResNet transformed “deep” from aspiration to a practical and efficient reality. Suddenly, depth was no longer symbolic. The experimental strategies presented in the original ResNet paper reflect Sutskever’s belief that progress in AI emerges from refining known ideas and uncovering what genuinely works. The subsequent ResNet v2 paper also embodies this ethos: progress through iteration, not invention.

3.1 Telephone Game

Early influential architectures gradually increased their depth, beginning with AlexNet’s modest eight layers in 2012, followed by VGGNet in 2014 with nineteen layers (VGG-19), and GoogLeNet, also introduced in 2014, with twenty-two layers. Yet, all attempts to train even deeper networks encountered obstacles. Chief among these were the persistent vanishing gradient and degradation problem.[\[2\]](#) Despite partial solutions such as batch normalization and careful weight initialization to stabilize gradients and assuage vanishing gradients and gradient clipping to limit exploding gradients, no comprehensive solution had emerged.

Imagine passing information through a series of layers, each designed to extract patterns from the original data. Initially, each of these layers is assigned random weights at random positions. As information, such as an image, passes through each layer, these random weights slightly distort the data. This distortion is manageable with only a few layers, but these subtle distortions compound as the network increases in depth. When information reaches the final layers of a multi-layer network, the original image or data has effectively been scrambled into random noise, losing nearly all meaningful connection to what was initially fed into the system. This deterioration of meaningful signals is known as the *degradation problem*.

After forward propagation, the network measures performance by calculating the difference between its distorted output and the correct answer. This difference is called the “loss.” To improve, the network propagates the loss backward through its layers to determine how each layer’s weights should be adjusted. As the feedback signals (i.e., gradients) travel backward through each layer, they become progressively smaller and distorted due to repeated multiplications by the layers’ weights. By the time these gradients reach the earliest layers, those closest to the original input, they are diminished to the point that they carry no meaningful information, making it impossible for these early layers to update their weights effectively. This phenomenon, where gradients effectively disappear as they propagate backward, is known as the *vanishing gradient problem*.

Although the degradation and vanishing gradient problems are distinct, they reinforce each other in a harmful feedback loop like two people arguing until the original point is either forgotten or completely distorted. The distorted forward signals corrupted the output, leading to weaker gradients during backpropagation; weaker gradients resulted in poor learning, contributing to distortion in subsequent forward passes, trapping researchers in a frustrating loop of diminishing returns. This vicious cycle made training deep networks extremely slow and challenging.

Against this backdrop, Microsoft researchers introduced ResNet. In their seminal paper “Deep Residual Learning for Image Recognition,” the authors addressed longstanding issues through a deceptively simple architectural innovation: residual connections. Imagine playing the “telephone game,” where a message is whispered from person to person down a line. Typically, when the message reaches the last person, it is distorted because each participant slightly mishears or misinterprets it. Now, imagine if you were allowed, at certain points, to pass a note directly to someone further along, altogether skipping multiple intermediate whisperers. This shortcut would ensure that at least part of the original message remains clear and intact.

This is precisely the idea behind residual connections.

Rather than forcing every signal to pass sequentially through many layers, where distortions accumulate, residual connections create shortcuts, allowing information to flow both *through* and *around* certain groups of layers. This dual path structure ensures meaningful information reaches deeper layers intact, and clear gradient signals flow smoothly back through the network.

This simple architectural design has significant implications for model training. During forward propagation, each residual block computes two parallel paths: one transforms the input through convolutional layers, batch normalization, and nonlinear activations to extract complex features, while the other directly forwards the original input unchanged, preserving the initial information. During backpropagation, gradients that determine weight updates travel backward along these same two paths. The direct shortcut helps maintain stronger gradients thus reducing the risk of vanishing gradients as they propagate backward and improving the stability and effectiveness of training deeper networks.

ResNet was a step function in computer vision performance. The architecture swept first place in every category (e.g., classification, localization, and detection) at the 2015 ImageNet Large Scale Visual Recognition Challenge (ILSVRC) and won in the detection and segmentation categories at the COCO 2015 challenge.^[3] Remarkably, an ensemble of ResNets achieved a top-5 error rate of just 3.57% on ImageNet.^[4] Even a single ResNet-152 model achieved a 4.49% top-5 error, surpassing previous results only attainable through computationally expensive ensembles of multiple models. This achievement earned ResNet the prestigious 2016 Best Paper Award at the Conference on Computer Vision and Pattern Recognition (CVPR), underscoring its impact on a genuinely shocked community. Reflecting on their success, Kaiming He remarked that ResNet “destroys some of the assumptions” he had previously held about the limitations of deep learning.^[5]

Residual connections are a special type of skip or “shortcut” connection. Residual connections allow earlier layers to connect directly to deeper layers, bypassing intermediate transformations. Specifically, each residual block computes an incremental adjustment called the “residual” through its internal layers, then adds this result directly to the original input. Formally, the output of each residual block is:

$$= () +$$

Here, $()$ represents the output produced by the block’s internal transformations, while x represents the original input. See Figure 3.1. By explicitly adding the original input, residual connections preserve clear signals during forward propagation and stabilize gradient flow during backward propagation.

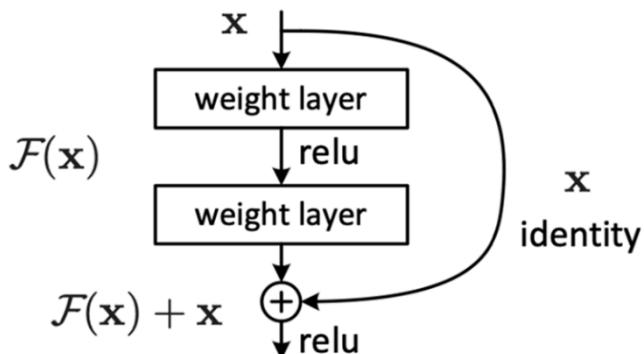


Figure 3.1 A diagram showing a simple ResNet block. It illustrates how input data flows along two paths: one *through* two layers of transformations with activation steps, and another *around* using a direct “shortcut” path without any transformations. The outputs of these two paths are *combined*, and the combined result goes through a final activation step. This “shortcut” helps the network learn more effectively by preserving information from earlier layers. *He et al. (2015), Fig. 2.* Used with explicit permission granted by the lead author (Kaiming He).

Network architectures without skip connections struggled at increased depths. Figure 3.2 illustrates this by showing that a “plain” 56-layer network without residual connections produced higher training errors than a shallower 20-layer network also without residual connections. The deeper 56-layer network shows higher training error than the shallower 20-layer network, indicating difficulties in optimization without additional architectural innovations. The deeper network also has higher test error, highlighting that adding more layers doesn’t necessarily improve performance. The unexpected increase in training errors wasn’t due to overfitting but revealed a fundamental difficulty in training deep networks.

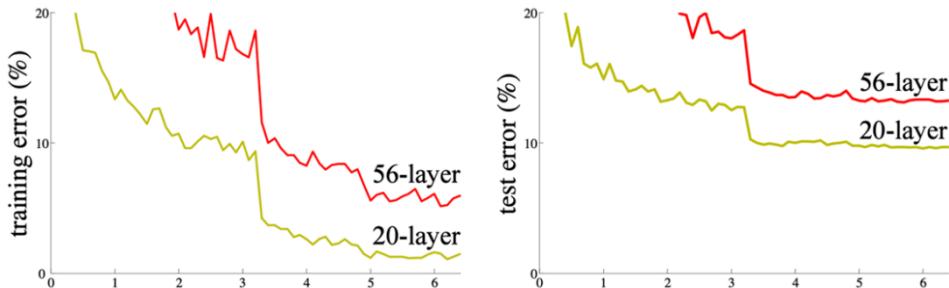


Figure 3.2 Training error (left) and test error (right) on CIFAR-10 with 20-layer and 56-layer “plain” networks. The deeper 56-layer network shows higher training error than the shallower 20-layer network, indicating difficulties in optimization without additional architectural innovations. The deeper network also has higher test error, highlighting that adding more layers doesn’t necessarily improve performance. The x-axis in both plots denotes training iterations in units of 10,000, with each iteration corresponding to a single mini-batch update. As training progresses across approximately 60,000 iterations, the deeper network consistently underperforms, emphasizing that depth alone is not sufficient for effective learning. *He et al. (2015), Fig. 1.* Used with explicit permission granted by the lead author (Kaiming He).

However, when residual connections are introduced, deeper networks outperform “plain” counterparts. The authors compare two plain networks without residual connections: one has 18 layers, and the other is a deeper version with 34 layers (Figure 3.3). Intuitively, one might expect the deeper 34-layer network to perform better, or at least no worse, since the deeper network should, in principle, represent all the functions the shallower 18-layer network can. That is, the solution space of the 18-layer network is essentially a subset of the deeper 34-layer model. Yet, surprisingly, the opposite happens. The deeper network performs worse (i.e., higher training and validation error) throughout the entire training procedure. This counterintuitive outcome clearly illustrates the “degradation problem.”

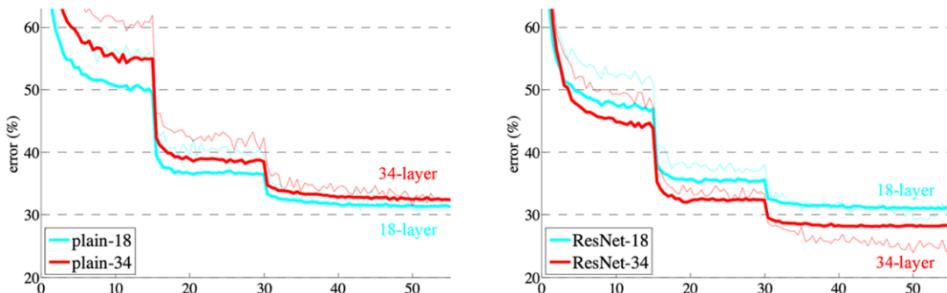


Figure 3.3 Thin error curves indicate training error on ImageNet, while bold error curves indicate validation error on ImageNet. Left: plain networks with 18 and 34 layers. Right: ResNets with 18 and 34 layers. The residual networks have no additional parameters compared to their plain counterparts. He et al. (2015), Fig. 4. Used with explicit permission granted by the lead author (Kaiming He).

The challenge for networks without residual connections is that each layer must learn a useful transformation. However, forcing layers to learn transformations even when unnecessary can destabilize training. Residual connections resolve this by allowing layers to pass inputs forward unchanged, known as the identity mapping, whenever no beneficial transformation is found. If a transformation is needed, layers learn small incremental adjustments known as *residuals* relative to the input. Thus, additional layers become safe to add because residual connections do nothing by default, preserving performance rather than harming it, and when needed, allowing targeted, beneficial modifications that gradually refine representations.

This behavior acts as a form of regularization since layers are encouraged to do nothing unless a more complex transformation improves performance. Therefore, residual networks can achieve greater depths without a corresponding rise in computational costs, unlike traditional architectures that face substantial computational penalties with each additional layer. For instance, ResNet-152 is approximately eight times deeper than the VGG-19 architecture yet requires fewer computational operations (FLOPs) and has a similar parameter count. The 152-layer ResNet has 11.3 billion FLOPs, lower than VGG-16/19, with 15.3 and 19.6 billion FLOPs, respectively. ResNet-34 requires only 3.6 billion FLOPs, approximately 18% of VGG-19's 19.6 billion FLOPs, even though ResNet-34 has almost twice as many layers as VGG-19. This efficiency is due to residual connections and, ResNet's specialized "bottleneck" design.

To understand how these bottlenecks work, it helps to consider the role of *channels* in convolutional neural networks. In an image, channels correspond to dimensions like red, green, and blue, each capturing different aspects of the same visual input. You can think of them as different camera angles recording the same scene from slightly different perspectives. Bottleneck layers reduce and expand these channels in a controlled way, compressing the computational burden while preserving expressive capacity.

Initially, multiple channels (or camera angles) help the network gather comprehensive information. However, processing every channel thoroughly is computationally expensive and often unnecessary. ResNet's bottleneck blocks cleverly address this problem. First, a 1×1 convolution reduces or "compresses" the number of channels and discards redundant information, much like a film editor selects the most useful camera angles. Next, a 3×3 convolution processes only this simplified representation. Finally, another 1×1 convolution restores or "expands" the channels back to their original dimensions. This compress-process-expand strategy reduces complexity compared to directly stacking larger convolutions, enabling ResNet to achieve greater efficiency without sacrificing performance.

Think of the bottleneck as a data compressor that first shrinks the information, processes this compressed version, and then expands the information back to its original form. Since the core calculations occur on smaller amounts of data, ResNet could perform better without additional computational resources. This meant that ResNet could have more layers. Indeed, just three years earlier, AlexNet halved the top-5 error rate on ImageNet with only eight layers. However, halving the error rate again required significantly more layers as performance approached saturation. ResNet halved the top-5 error rate and demonstrated that extraordinary depth was possible, including an unprecedented 1,202-layer network.

Yet, ResNet's depth attracted its share of criticism. While deeper networks became feasible, there was no principled method for determining optimal depth. The original ResNet paper empirically found 152 layers optimal for ImageNet yet offered no theoretical or analytical framework to guide depth selection. Practitioners had little choice but to rely on trial-and-error. Moreover, ultra-deep ResNets still demanded substantial computational resources and large datasets; otherwise, they risked overfitting. Subsequent architectures, including ResNeXt, DenseNet, ShuffleNet, and EfficientNet, reinforced these concerns by demonstrating that carefully designed models could match or exceed ResNet performance using significantly fewer parameters and less data.[\[6\]](#)[\[7\]](#)[\[8\]](#)[\[9\]](#) Nevertheless, ResNet fundamentally redefined what was possible.

3.2 Comparison with Other Major Architectures

By examining how previous architectures tackled network depth, complexity, efficiency, and representational capacity, we can better understand the significance of ResNet's innovations and their impact. Below are comparisons of their design philosophies, performance, and efficiency, which emphasize key differences:

3.2.1 ResNet vs. AlexNet & ZFNet

AlexNet introduced the ReLU activation function, dropout for regularization, and GPU-based training. In 2013, ZFNet refined AlexNet by adjusting its convolutional layer to use smaller (7×7) convolutional filters instead of AlexNet's larger (11×11) filters. This slightly enhanced performance, which confirmed the importance of careful architectural tuning. Yet, despite their achievements, AlexNet and ZFNet quickly encountered limitations due to persistent optimization issues. Consequently, AlexNet and ZFNet were unable to surpass about a dozen layers without facing training instability or decreased performance.

ResNet is on *Sutskever's List* because it represents a pivotal shift in design philosophy. Unlike AlexNet or ZFNet, which were breakthroughs but manually tuned architectures, ResNet introduced a fundamentally new approach: a modular, repeatable residual block that drastically simplified the scaling of artificial neural networks. The idea that stacking a standard module n times can outperform a heterogeneous design is formidable. This simplicity is precisely the fundamental innovation that characterizes many entries on *Sutskever's List*.

3.2.2 ResNet vs. VGGNet

Following AlexNet and ZFNet, researchers at Oxford University's Visual Geometry Group introduced VGGNet in 2014. Instead of carefully engineering each layer, VGGNet stacked identical convolutional layers. Philosophically, this approach anticipated ResNet more closely than AlexNet or ZFNet. VGGNet primarily used uniform 3×3 convolutions interspersed with pooling and demonstrated that consistently repeating a small building block to gain depth could deliver significant performance gains. Their 16-layer VGG-16 variant achieved a 7.3% top-5 error rate on ImageNet.

Yet, despite sharing this modular philosophy with ResNet, VGGNet encountered practical limitations. Its large, fully connected layers resulted in an enormous parameter count of over 138 million and significant computational demands during training and deployment. Moreover, stacking layers beyond 19 still led to unstable training due to the vanishing gradient problem. Scaling networks was not yet possible without an architectural innovation to keep signals alive.

3.2.3 ResNet vs. GoogLeNet

In 2014, Google introduced GoogLeNet.[\[10\]](#) Rather than stacking identical convolutional layers or relying on uniform building blocks, GoogLeNet featured carefully engineered "Inception modules." An Inception module acts as a miniature network that simultaneously processes input features through multiple parallel paths. Different convolutional filter sizes (typically 1×1 , 3×3 , and 5×5) and pooling operations run concurrently within a single module. The outputs from these parallel paths are merged, allowing the network to capture features at multiple scales within a single layer. GoogLeNet achieved an impressive top-5 error rate of around 6.7% on ImageNet with just 22 layers and roughly 6.8 million parameters.

Yet, despite its effectiveness, the Inception architecture introduced complexity. Each Inception module required manual tuning and balancing of multiple parallel computational paths. While GoogLeNet was clever and achieved impressive accuracy with fewer parameters through its hand-crafted Inception modules, its design was brittle and hard to scale. Each new version required manually tuned filter sizes and parallel branches tailored to specific depths.

In contrast, ResNet embraced a more straightforward strategy. Rather than intricate multi-path modules, ResNet repeated a standardized residual block multiple times. This allowed networks to scale (e.g., ResNet-18, -34, -50, -101, -152, -1202) without redesign. It solved the vanishing gradient problem and became the default backbone for countless downstream models. Even Inception eventually conceded, adopting residual connections in Inception-ResNet (2016).[\[11\]](#) The table below summarizes some key metrics of VGG, Inception, and ResNet. It highlights the dramatic increase in depth and improvement in accuracy.

Table 3.1 This table compares VGG-19, GoogLeNet (Inception v1), and ResNet-152.

Architecture	Year	Depth (layers)	Parameters	ImageNet Top-5 Accuracy
VGG-19	2014	16 conv + 3 FC	~138 million	~92.7%
GoogLeNet (Inception v1)	2014	22	~6.8 million	~93.3%
ResNet-152	2015	152	~60 million	~95.5%

3.3 Real-World Adoption

While some argued that ResNet's core idea was not entirely novel, its dominance persisted beyond the initial 2015 results.[\[12\]](#) Models like ResNeXt-101 (2016) and SENet (2017) secured top positions in the ImageNet competitions.[\[13\]](#)[\[14\]](#) Researchers quickly transitioned from older architectures, such as VGG-based models, to ResNet variants, observing immediate and substantial performance gains. The winning formula in computer vision had become "ResNet plus some tweaks." By 2017, nearly all ImageNet entries cited ResNet as their backbone. For example, replacing VGG-16 with ResNet-101 for Faster R-CNN yielded a remarkable 6–7% accuracy increase on the COCO dataset.[\[15\]](#)

Residual connections have become foundational in various real-world computer vision applications. A 2023 survey highlighted their extensive deployment in clinical decision support systems, where they assist in diagnosing lung tumors from CT scans, detecting breast cancer from mammograms, and classifying skin lesions.[\[16\]](#) ResNets are prized for their robustness in scaling and viewpoint variations in autonomous driving. Tesla's Autopilot system, for instance, reportedly uses a modified ResNet-50 backbone referred to as "HydraNet" to process multi-camera input for tasks such as object detection, lane recognition, and depth estimation.[\[17\]](#)[\[18\]](#) Research consistently shows that deeper variants like ResNet-101 outperform shallower models in particularly challenging tasks like lane detection.[\[19\]](#)

Adoption spread across the tech industry. Google incorporated them into hybrid architectures like Inception-ResNet.[\[20\]](#) Facebook followed suit with frameworks such as Mask R-CNN and ResNeXt while Amazon introduced ResNeSt.[\[21\]](#)[\[22\]](#)[\[23\]](#) ResNet became the default backbone for production-grade image recognition across platforms like Google Photos, Pinterest, and Microsoft Azure's cognitive services. By 2018, widespread adoption in platforms like NVIDIA's Detectron and PyTorch's torchvision provided reliable implementations into production pipelines. Its versatility, efficiency, and strong performance on large-scale vision tasks made ResNet the de facto starting point for industry-scale computer vision deployments and remains widely used today.

3.4 ResNet-v2

While the original ResNet architecture enabled the training of very deep networks through residual connections, it wasn't without flaws. Researchers found that placing nonlinear activations *after* adding the skip connection could still disrupt signal propagation, especially in deeper networks. This insight led the same authors to revisit their design in the follow-up paper "*Identity Mappings in Deep Residual Networks*," introducing ResNet v2.

The core innovation in ResNet v2 was again deceptively simple. Reordering operations within each residual block preserves the integrity of the skip connection. In the original design, each block combined a transformed path (convolutions + activations) with an identity path and then applied a final ReLU activation. ResNet v2 eliminated this post-merge activation and pushed all normalization and activation steps *before* the convolutions. This subtle shift made the skip connection a clean, unaltered identity mapping, improving gradient flow and simplifying optimization, especially in ultra-deep architectures.

An intuitive analogy may help clarify this idea: Imagine residual connections as a freeway designed to quickly move traffic (or information) around a city (the network). The original ResNet design placed a traffic light (activation function) right after merging two busy roads (the skip connection and convolutional path). Needless to say, this would cause even more congestion, especially when the “traffic light” unexpectedly turns red, like when an activation function saturates. Recognizing this problem, researchers redesigned the intersection in ResNet v2. They shifted all traffic controls onto the convolutional path. Consequently, the traffic is free to move around the city, thus avoiding downtown traffic. This adjustment ensures information flows effortlessly through and around the network, free from bottlenecks and traffic lights; something all commuters would envy during rush hour.

3.4.1 Post-Activation to Pre-Activation

The original ResNet architecture placed convolutional layers first and applied activation functions last within each residual block. A basic ResNet block operated in this sequence: convolution batch normalization ReLU convolution batch normalization, followed by adding the skip connection. After combining these two paths, a final ReLU activation was applied to the result.

In contrast, the improved “pre-activation” ResNet v2 rearranged operations within residual blocks. In this new design, each block applies batch normalization and ReLU activation before each convolutional layer rather than afterward, and crucially, no activation function is used after merging with the skip connection. Thus, the sequence becomes: input batch normalization ReLU convolution batch normalization ReLU convolution addition of skip connection. See Figure 3.4.

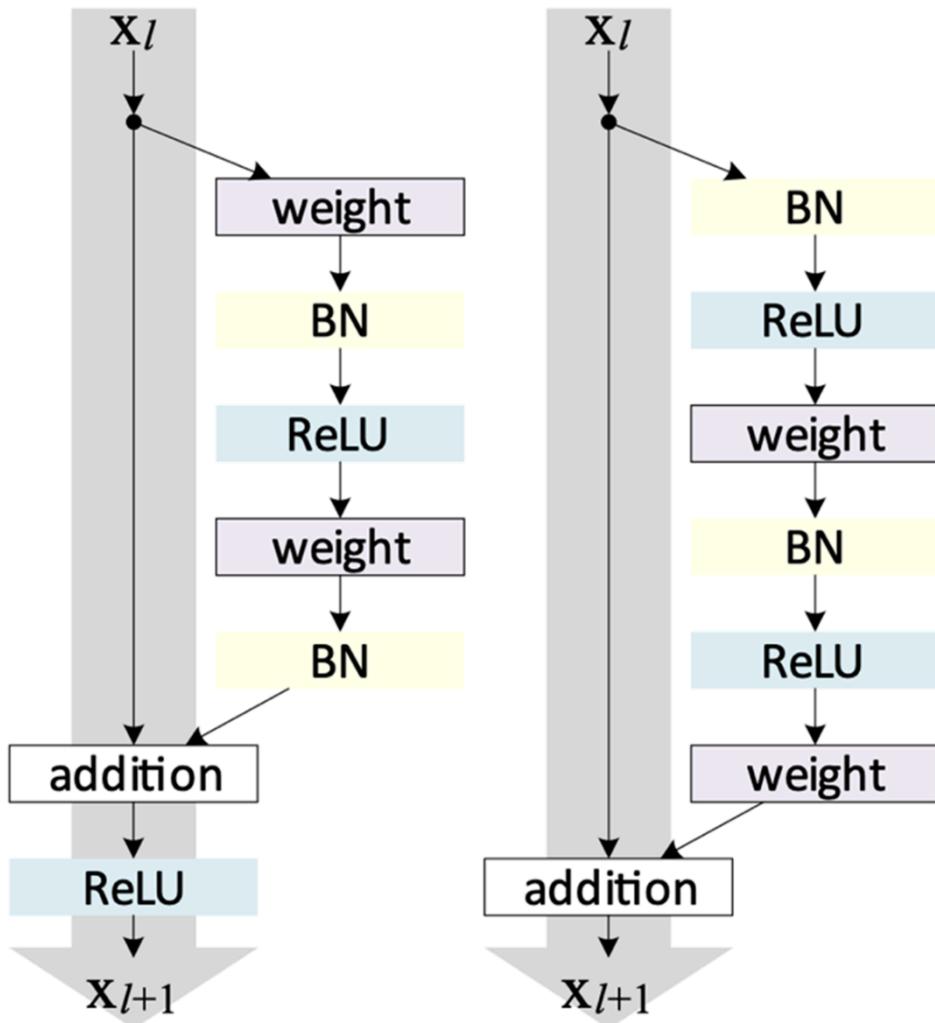


Figure 3.4 (Left) The original residual unit shows two weight layers with Batch Normalization (BN) and ReLU activations in between, followed by an additional step merging back the identity connection. **(Right)** The revised residual unit reorganizes operations by placing BN and ReLU before the weight layers, simplifying the pathway. The grey arrows highlight the direct paths facilitating easy forward and backward flow of information through the residual (shortcut) connections. *He et al. (2015), Fig. 1 (left).* Used with explicit permission granted by the lead author (Kaiming He).

Immediately after combining paths, the output passes directly into the next residual block without further processing, maintaining a pure, additive identity mapping. By removing the nonlinear activation (ReLU) after addition and confining all nonlinearities and normalization within the convolutional path, the skip connection becomes a transparent shortcut. This creates a clean information pathway that allows data and gradients to flow smoothly across many layers. This ensures stable training at increased depth.

Additionally, placing batch normalization (BN) at the start of each convolutional layer provides normalized inputs, introducing regularization benefits that improve generalization performance. Indeed, the authors observed that pre-activation ResNets had slightly higher training errors but lower test errors than their original post-activation counterparts, a phenomenon attributed to the regularizing effect introduced by applying batch normalization at the input of each convolutional layer.

For readers seeking a formal understanding of why activation placement matters, consider the following explanation: In the original ResNet design, the ReLU activation is applied after adding the residual branch and skip connection outputs. As a result, the activation modifies both paths simultaneously. In contrast, the pre-activation design applies activation functions asymmetrically. That is, only the residual branch undergoes the nonlinear ReLU activation, while the skip connection remains unchanged. By applying ReLU only within the residual path, the skip connection maintains its “pure identity,” ensuring the combined output experiences no nonlinearity after addition. This asymmetry creates the ideal conditions to facilitate the uninterrupted information propagation through the network.

3.4.2 Ablation Studies

Ablation studies systematically remove or modify parts of a model to determine their individual contributions. The authors used ablation studies to systematically test several modifications to the skip connection to validate the importance of preserving identity mappings. Even minor changes, such as scaling the skip path by a constant (e.g., 0.5) or randomly dropping it during training, led to degraded performance and poor convergence. More sophisticated tweaks, including learnable gates or adding convolutional layers to the skip path, fared even worse, often increasing error rates or causing optimization to fail entirely. At these depths, minor architectural decisions became make-or-break. The takeaway was clear: the simplest, unaltered identity skip connection was more elegant and far more effective.

The authors also evaluated how changing the position of operations, such as activation functions (ReLU) and normalization steps (batch normalization), relative to the skip connection affected performance. Compared to the original approach of placing activations after combining paths, they found that putting batch normalization directly after the addition disrupted the skip path and harmed accuracy. Similarly, applying ReLU before addition restricted the residual branch to positive adjustments only, limiting effectiveness. Using ReLU alone before each convolution without batch normalization did not meaningfully improve performance either. Ultimately, the most effective approach was the full pre-activation strategy, which included placing batch normalization and ReLU before convolutions and omitting activations after the addition.

The insistence on preserving pure identity mappings in ResNet v2 reflects a broader philosophy that values empirical clarity over theoretical novelty, which aligns closely with Ilya Sutskever's approach to research. Rather than inventing entirely new mechanisms, the authors rigorously tested existing assumptions through systematic experimentation, revealing that even subtle deviations, like scaling or gating the skip connection, could derail training. This experimental mindset mirrors Sutskever's belief that genuine progress in AI emerges not from chasing conceptual breakthroughs, but from deeply understanding known ideas, refining them at scale, and uncovering what actually works. ResNet v2 paper embodies this ethos that resonates with Sutskever: progress through iteration, not invention.

Armed with pre-activation residual units, the authors successfully trained an ultra-deep 1,001-layer ResNet on the CIFAR-10 dataset. This ultra-deep network achieved record-setting accuracy, reducing classification error to 4.62%. In comparison, a 1,202-layer original ResNet architecture had an error rate of 7.93%. At a time when training deep networks was still an ordeal, the authors demonstrated that extremely deep architectures could be stable if carefully designed. The takeaway was clear: depth was no longer the enemy, so long as the architecture respected the constraints imposed by optimization.

On the much larger ImageNet dataset, the authors successfully trained a 200-layer pre-activation ResNet that exceeded the accuracy of the original 152-layer post-activation version. What's most striking is the razor-thin margin for success. Extending the original post-activation design to 200 layers caused performance to degrade, not because the model was too deep in principle, but because the architecture couldn't support stable optimization at that scale. This wasn't merely a subtle improvement; it reflected the difference between model convergence and model collapse.

Subsequent theoretical work provided insight into why this design succeeded. Research showed that ResNets effectively behave like ensembles of many shallower networks of varying depths.[\[24\]](#) Gradient propagation in very deep networks often occurs through shorter, more direct pathways rather than traversing every single layer sequentially. Thus, a 1,000-layer ResNet isn't truly functioning as a single chain of 1,000 sequential layers; instead, it includes numerous shorter pathways that gradients and signals can use. These shorter pathways help avoid vanishing gradients, making training significantly more stable. Therefore, additional layers can be added without harming performance because unused layers can be bypassed. The identity mappings effectively allowed the network to select and utilize layers as needed, ignoring non-helpful ones. This made exploring extreme networks as deep as 1,001 layers feasible, unlocking new possibilities for scaling.

The authors supported this idea through what they charmingly called "lesion studies," a term borrowed from neuroscience, presumably to lend an air of surgical precision to what everyone else in machine learning calls "ablation studies." In these "lesion studies," they systematically dropped out individual layers to observe the impact, though perhaps the authors genuinely imagined themselves surgically extracting layers from the network with a programmatic scalpel in hand.

They found that dropping individual layers using residual connections caused only a modest increase in error, rather than catastrophic failure. This suggests that the network can compensate by using other available paths. Their analysis showed that in a 110-layer ResNet, most effective gradient flow occurred along shorter subnetworks (approximately 10–34 layers), whereas very deep paths contributed minimally, essentially learning identity mappings.

Thus, ResNet effectively behaves like an ensemble of shallower networks during training. Each residual block can contribute features or be bypassed, resulting in a network whose predictions reflect an averaging or combination of multiple shorter models. This ensemble perspective explains how ResNet can achieve extremely deep structures without severe overfitting or optimization difficulties. This theoretical interpretation reveals that ResNet's success arises not merely from having one very deep model, but from combining many simpler, overlapping models facilitated by residual connections.

By 2025, purely convolutional networks will have been complemented or partially replaced by transformer-based or hybrid architectures, particularly for tasks requiring complex long-range dependencies modeling. Yet even these advanced models rely on residual connections for stable, practical training. The core ResNet philosophy remains highly influential: preserve what's working and only learn what's necessary. This principle essentially encourages each layer to perform minimal, incremental adjustments ("residuals") rather than completely transforming the data. These minor refinements accumulate through the layers, progressively enhancing the representation. This incremental approach continues to influence modern architectural design in computer vision and beyond. Given its foundational impact on how extremely deep neural networks are optimized and trained, it's no wonder this paper earned a place on *Sutskever's List*.

3.5 Extending ResNet with Dense Prediction

ResNet was designed for image classification, where the goal is to assign a single label to an image. For image classification, mechanisms like pooling, which summarize local regions into single values, and strided convolutions, which skip pixels to downsample the image, are highly effective. These methods reduce spatial resolution, aggregating information and effectively discarding pixel-level details. However, these techniques hinder dense prediction tasks like semantic segmentation, where high spatial resolution is essential for precise pixel-level labeling and sharp boundary delineation.

One might avoid pooling altogether and instead enlarge convolutional kernels to increase the receptive field, the region of the input each neuron "sees." After all, a large receptive field can capture contextual information across wide image regions, which is essential in dense tasks like semantic segmentation. However, increasing kernel sizes is impractical. For example, enlarging a modest 3×3 kernel (nine parameters) to a 15×15 kernel (225 parameters) dramatically inflates parameter count, computational complexity, and memory usage. Moreover, large kernels fail to capture fine-grained details, blurring precise local features critical for pixel-level accuracy. Thus, traditional architectures such as AlexNet and VGGNet rely on small kernels combined with pooling and stacking multiple layers to incrementally grow receptive fields without excessively increasing model complexity. Yet even this approach faces limits for dense prediction tasks. The influential 2016 paper "Multi-scale Context Aggregation by Dilated Convolutions" proposed an alternative strategy to address these trade-offs.[\[25\]](#) The paper, featured on *Sutskever's List*, introduced dilated convolutions.

Unlike standard convolutions, dilated convolutions insert gaps between filter elements. Instead of increasing the kernel size itself, dilated convolutions strategically skip pixels according to a specified dilation factor. This mechanism enables the model to integrate multi-scale context and capture longer-range dependencies while maintaining full spatial resolution without excessive parameter surge. Think of it as looking at the same picture through powerful magnifying glasses at different scales. Each successive layer offers a wider field of view while maintaining clarity and detail. Rather than using filters, dilated convolutions skip pixels proportionally to their dilation. The receptive field grows dramatically by exponentially increasing this dilation factor (e.g., 1, 2, 4, 8). In contrast, the number of parameters grows only linearly.

This innovation lies at the heart of the **context module**, a component explicitly designed for multi-scale context aggregation. By stacking dilated convolutions with increasing dilation rates, the context module enables the network to capture fine local structure and broad global dependencies. It addresses a core limitation of traditional ResNet-like architectures in dense prediction tasks, which often trade spatial resolution for deeper abstraction. The context module sidesteps that compromise, enabling expansive context to be gathered without sacrificing pixel-level detail or computational efficiency. This strategy, illustrated in Figure 3.5, enables efficient multi-scale context aggregation at full resolution, capturing expansive contextual information with remarkable parameter efficiency.

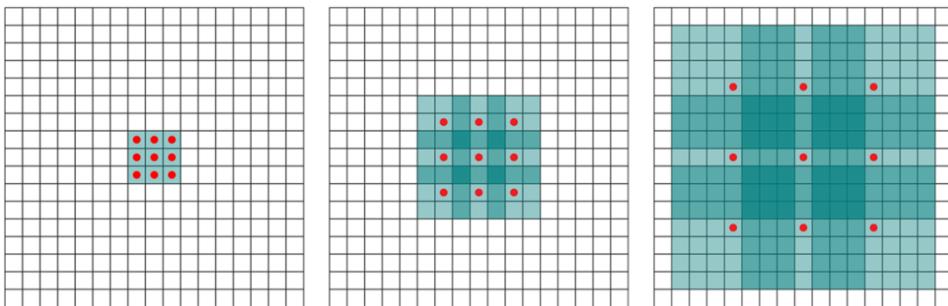


Figure 3.5 Dilated convolutions rapidly expand the receptive field without increasing computational cost. Each red dot represents a convolutional kernel element. As dilation increases (from left to right), the receptive field (blue-shaded area) expands exponentially, capturing broader spatial context while maintaining the same number of parameters. Yu and Koltun (2016), Fig. 1. Used with explicit permission granted by the lead author (Fisher Yu).

To complement this mechanism, the authors streamlined their front-end architecture by adapting the well-known VGG-16 backbone. Recognizing that dense pixel labeling requires high-resolution feature maps, the authors removed the last two pooling and striding layers, which are traditionally used for downsampling in classification tasks. Instead, they utilized dilation in subsequent convolutions. By stripping away classification-centric design elements, the authors created a simpler, purpose-built model tailored for semantic segmentation. The modified front-end achieved higher accuracy with an improvement of more than five percentage points over other models on the PASCAL VOC benchmark while also reducing architectural complexity and computational overhead.

Yet training specialized networks with dilated convolutions for dense prediction proved challenging. Conventional random weight initialization repeatedly failed. The breakthrough came from an old friend: the identity function. Rather than initializing layers with random weights, the authors proposed **identity initialization**, allowing layers to pass inputs forward unchanged. The solution felt almost too simple, but that was the point.

Like the ResNet skip connection, which only intervenes when it improves the signal, identity initialization starts each layer in a stable, no-operation state. In this configuration, the layer does nothing and passes its input forward unchanged. By behaving like an identity function, each layer avoids introducing noise or distortion into the network's early computations. This minimalist starting point preserves the signal and allows the model to learn only the deviations that improve performance. Empirical results validate the approach. Tested on the Pascal VOC 2012 dataset, the context module significantly outperformed previous state-of-the-art models, surpassing prominent semantic segmentation methods such as FCN-8, which replaced fully connected layers with convolutional layers to predict pixel-wise labels, and DeepLab, which employed conditional random fields to improve segmentation accuracy by over five percentage points.[\[26\]](#)[\[27\]](#)

Residual connections implement this by introducing skip connections, allowing information to flow both *through* and *around* specific groups of layers. Identity initialization achieves a similar effect by setting the network's initial weights so that each layer begins as an identity mapping, producing outputs nearly identical to inputs. At the start of training, identity initialization ensures layer outputs closely match their inputs. This simplifies optimization since the network needs only to learn incremental adjustments or "residuals" rather than complex transformations. This provides a stable starting point without external structural modifications. It's a ResNetian maneuver through and through: minimalist, robust, and surprisingly powerful.

Figure 3.6 visually compares the outcomes of different models in the specific semantic segmentation task, highlighting qualitative differences. The figure shows how accurately different models can identify and outline distinct objects in images. It showcases five unique scenes, including a person riding a horse, a fighter jet soaring through the sky, a bird mid-flight, furniture in a living room, and chairs arranged together. Each scene is presented in multiple versions: first, the original photo; then the predictions made by three different segmentation approaches; and finally, the actual correct outlines known as the ground truth. The methods vary in precision. The first method (FCN-8s) produces rough outlines but struggles to capture fine details. The second method (DeepLab) improves accuracy, outlining objects more closely but still with some imperfections. The authors' proposed method further refines these outlines, nearing the accuracy and crispness of the ground truth. Overall, the visual comparison emphasizes how dilated convolutions enhance dense prediction.

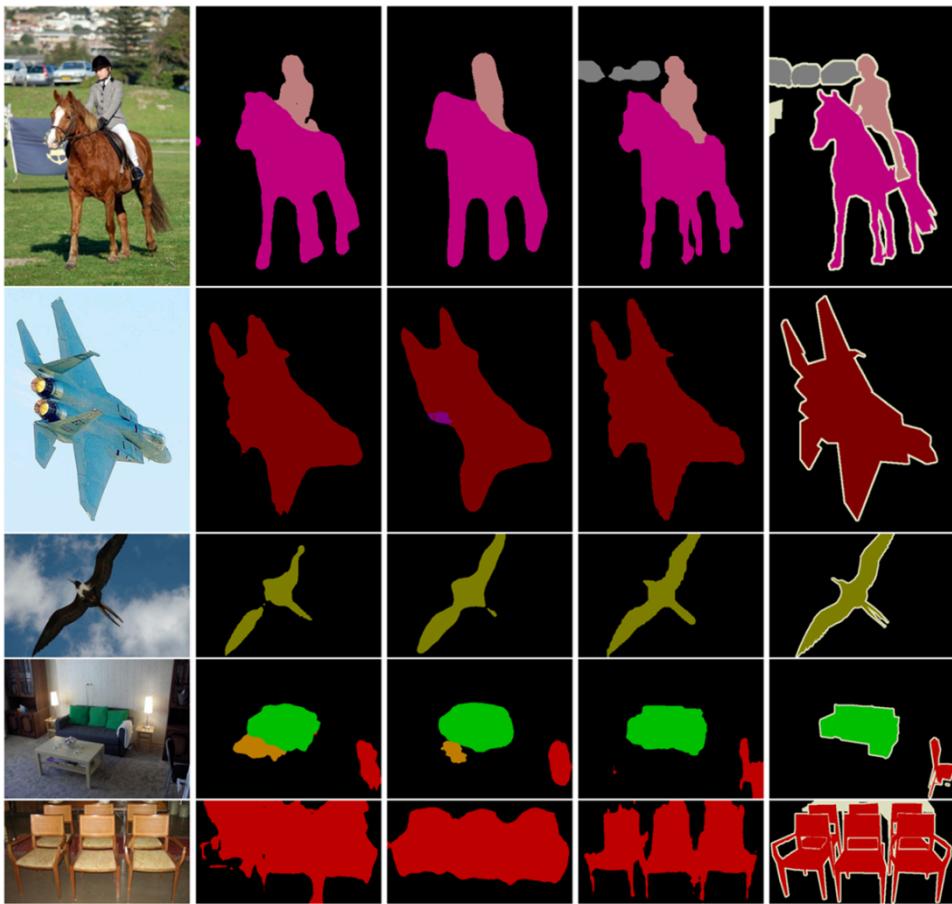


Figure 3.6 Semantic segmentations produced by different classification networks. From left to right: the first column is the input image, the second is the prediction by FCN-8s, the third is the prediction by DeepLab, the fourth is Yu and Koltun (2015), and the last is the ground truth. The examples illustrate progressively clearer and more accurate segmentations, with Yu and Koltun (2015) significantly improving boundary definitions and object identification compared to previous methods. Yu and Koltun (2015), Fig. 2. Used with explicit permission granted by the lead author (Fisher Yu).

While ResNet excelled at classification, its reliance on pooling and downsampling made it poorly suited for dense tasks like segmentation, where spatial precision is crucial. The breakthrough came with *dilated convolutions* that expand the receptive field without sacrificing resolution or increasing the number of parameters. This led to the creation of the *context module*, which captures both local details and global context at full resolution by stacking dilated convolutions with increasing dilation rates. Identity initialization further stabilized training by starting each layer in a no-operation state, where a layer does nothing and passes its input forward unchanged. Together, these design choices represent a pivotal moment in the evolution of deep learning for dense prediction.

3.6 Human Measuring Sticks

In a 2015 blog, Microsoft highlighted that their ResNet-powered systems could “meet and sometimes exceed human-level performance.”^[28] This achievement of surpassing a “human-level” benchmark energized public discussions about AI. This moment was different from AlexNet, which was essentially a private milestone celebrated among insiders and marked by bets and friendly wagers between leading researchers like Geoffrey Hinton and Jitendra Malik. In contrast, ResNet transitioned toward a period defined by explicit competition with human benchmarks and public displays.

The human threshold was introduced in late 2014 by Fei-Fei Li’s team at Stanford. It was derived from an extremely limited sample: two “expert annotators,” one trained on only 500 images and another on merely 100 images.^[29] This small experiment reported a human error rate of 5.1%, and 5% became entrenched as the threshold for human performance. Yet, concluding human-level performance from such a small dataset and just two annotators is methodologically questionable.

Yet methodology wasn’t the only concern. The validity of the benchmark itself was questionable. The ImageNet top-five metric was explicitly designed to measure machine performance, allowing predictions to count as correct if the true label appeared anywhere within the model’s top-five guesses. For example, if a model is presented with a basketball and returns guesses like “croquet ball,” “bikini,” “warthog,” “basketball,” and “moving van,” it still counts as correct.^[30] This permissive scoring is helpful in evaluating machines, where recognizing that the model is at least “close” provides diagnostic insight. However, retroactively applying this criterion to humans is awkward at best.

Humans rarely hedge among multiple guesses when distinguishing straightforward items like “basketballs” and “bikinis,” making the top-five metric conceptually incongruent as a measure of human performance. This post-hoc assignment introduced confusion about what “human-level” performance even meant. Besides, ResNet’s less frequently discussed top-one accuracy was 78.57 percent, significantly below the 95 percent threshold attributed to human performance.^{[31][32]}

The threshold effectively captured annotators’ proficiency at labeling images according to ImageNet’s specialized conventions rather than evaluating general human visual acuity. Such labeling involves idiosyncratic distinctions, like fine-grained breeds of dogs or birds, and subtle category boundaries tailored for training and evaluating machine learning models rather than reflecting general human perception in real-world scenarios.

Remember, ImageNet involves 1,000 object categories, including many challenging fine-grained classes. While humans excel at recognizing familiar objects like bikinis, they struggle with artificial categories, subtle distinctions, or unfamiliar specialized taxonomies, such as precisely distinguishing closely related dog breeds or rare bird species defined by the ImageNet dataset.^[33] The ImageNet team admits this in their 2014 paper, describing “annotating images with one of 1,000 categories as an extremely challenging task for an untrained annotator.”^[34] Thus, the human performance threshold offers limited insight into actual human performance. Instead, it indicates how well annotators adapt to the labeling conventions of academic benchmark datasets.

Nevertheless, the human-level threshold became influential as an external threshold for measuring progress in artificial intelligence, explicitly linking machine performance to human cognitive capabilities. In fact, ResNet's significance partly derives from surpassing this arbitrary and somewhat dubious yet symbolically powerful threshold, transforming a simple metric into a milestone many perceived as tangible progress toward "real" artificial intelligence, however vaguely defined that might be. This symbolism quickly escaped the confines of academic benchmarking and took on a life of its own in the public imagination, becoming media catnip.

It was frequently cited as proof that AI had become "better than humans" at vision. Headlines gushed with techno-triumphalism: EE Times ran "Microsoft, Google Beat Humans at Image Recognition" (Feb 2015); *Popular Mechanics* followed with "Microsoft Says Its Image Recognition Software Can Outperform Humans."^[35]^[36] *The Guardian* went further in May 2015: "Computers now better than humans at recognising and sorting images."^[37] By 2017, *Entrepreneur* asked, "Can Artificial Intelligence Identify Pictures Better than Humans?" The article was jauntily subtitled, "It's taken computers less than a century to learn what it took humans 540 million years to know."^[38] Apparently, evolution needs to hire better engineers.

The breathless coverage rarely paused to ask whether the benchmark captured human vision, or merely rewarded machines for out-labeling humans on a taxonomy no human cares about. Yet virtually none paused to consider an even deeper question: Even if the benchmark did accurately reflect human-level visual acuity, and that itself is debatable, would surpassing this threshold imply cognitive or psychological plausibility for the machine? Did "beating humans" on ImageNet indicate anything fundamental about how machines perceive, understand, or reason about the visual world, or merely confirm that they had mastered a labeling taxonomy irrelevant to genuine human vision?

In reality, surpassing the human-level threshold on ImageNet did not indicate human-like visual understanding or even general robustness. Indeed, convolutional neural networks (CNNs) would soon face criticism specifically for their lack of robustness and real-world generalization. Researchers discovered that these models were vulnerable to adversarial examples, inputs slightly perturbed in ways imperceptible to humans yet capable of fooling networks into confident misclassifications.^[39] This spurred interest in more interpretable model designs and post-hoc explanation techniques, because simply adding residual connections to CNNs did not address the lack of insight into *why* the models make confident decisions.

Robustness failures extended beyond synthetic adversarial examples, manifesting prominently in real-world deployments. For instance, in 2015, Google Photos mistakenly labeled images of Black individuals as "gorillas," illustrating how fragile model generalization to underrepresented examples can be despite the claims of human-level performance.^[40] Even as architectures grew deeper and approached human parity on benchmarks, they remained susceptible to errors no human would make when encountering subtle distribution shifts.

This discrepancy raised questions about what exactly constituted “human-level” performance if a model didn’t act like human intelligence. At the time, Google had already improved its original GoogLeNet architecture. Inception v3 achieved a 5.6% top-5 error on ImageNet, placing second only to ResNet’s 4.49% in the 2015 ILSVRC competition. Yet flirting with so-called “human-level” benchmarks did not immunize these models against real-world brittleness, highlighting the persistent gap between leaderboard metrics and genuine robustness.[\[41\]](#) Although depth offered superior representational learning compared to shallower models, achieving high accuracy on benchmarks alone provided few guarantees of robust generalization, particularly when training distributions were incomplete or when predictions spanned sensitive categories.

While ResNet did not establish this threshold, the Microsoft team behind it had embraced its symbolic weight. In an earlier, unrelated 2015 paper titled “Delving Deep into Rectifiers,” they wrote, “our result is the first to surpass human-level performance (5.1%) on this visual recognition challenge.” So, by the time the ResNet paper was released later that year, the team was already acutely aware of the rhetorical power of crossing the threshold and made that awareness explicit by citing the original human-level performance estimate in the paper summary. The 5% error rate would become more than just a metric, and ResNet crossed it with deliberate emphasis.

Despite its limitations, ImageNet’s 5% human benchmark was pivotal in modern artificial intelligence. It was pivotal, not because it was methodologically sound or valid, nor because it was the first time machines were compared to humans. There were instances of human measuring sticks, including IBM’s Deep Blue, the Chinook program, and IBM Watson’s victory in Jeopardy!, Facebook’s “DeepFace,” DeepMind’s Agent57, and AlphaGo. However, ImageNet’s threshold was one of the earliest fixed, quantitative human performance metrics to define and drive progress in the deep learning era.

The Microsoft research team leaned into this trend and helped to solidify “human-level performance” as an explicit research goal. This inspired others to follow suit, shifting the Overton Window in modern artificial intelligence to embrace human baselines as routine benchmarks. Today, whether it’s in vision, language, or speech, researchers commonly establish human baselines, create human-level thresholds, and celebrate models that reach or surpass them. While earlier achievements hinted at this shift, ResNet’s breakthrough popularized the now-common belief that human performance is a benchmark to beat.

From a philosophical perspective, reducing human capability to a numerical benchmark oversimplifies complex questions about intelligence and human cognition. Societally, treating humans merely as benchmarks to surpass shifts the emphasis away from creating complementary and beneficial technology toward competition, implicitly redefining humanity’s role from innovator to competitor. While AI may have been technically ready to chase benchmarks, most weren’t yet prepared for what it meant to surpass them.

3.7 ResNet Revolution

ResNet is a case study in problem-driven research. While vanishing gradients were well-known and model degradation had been observed, these issues weren't urgent until AlexNet, GoogLeNet, and VGGNet pushed model size beyond existing boundaries. Had ResNet been introduced earlier, it would still have worked, but its impact would have been trivial. This reflects a recurring pattern in AI research: while the field is often animated by abstract ambitions like achieving human-level general intelligence, progress comes from confronting specific failures. Unsurprisingly, problems need to be large enough to be seen before they can be solved which demands scale. Scale is not optional though it was once treated as a choice.

ResNet emerged naturally as a pragmatic response, and its effectiveness was amplified by the scale it enabled. This kind of flywheel effect magnifies the impact of minor architectural innovations, reinforcing incentives for greater scale. Ultimately, the underlying architecture is the core beneficiary, most visible at larger scales, precisely where artificial neural networks can harness increased representational power. The result is the closest thing machine learning has to a free lunch, provided innovations like residual connections and pre-activation enable stable training.[\[42\]](#)

In his influential 2019 essay, *The Bitter Lesson*, Richard Sutton argues that researchers are far better off by relying on general-purpose methods that scale with compute, rather than tuning hand-crafted heuristics or designing solutions based on how we think we think. We saw this dynamic play out in the previous chapter as AlexNet beat hand-crafted feature engineering. While encoding prior knowledge can offer short-term performance gains, these gains are eventually eclipsed by architectures that learn those features directly from data at scale.[\[43\]](#)

However, *The Bitter Lesson* ignores something important. As Ilya Sutskever repeatedly emphasizes in interviews, usually in response to *The Bitter Lesson*, we must clarify: "What exactly are we scaling?"[\[44\]](#) This omission by Sutton generates a sort of "Stone Soup" narrative. Both narratives emphasize that the real magic isn't the architecture and that researchers tend to give themselves too much credit. In the case of *Stone Soup*, data is the universal equalizer, and researchers should be more humble. *The Bitter Lesson* treats computation as the universal equalizer, and researchers should humble themselves. Ironically, researchers like Efros (*Stone Soup*) and Sutton (*The Bitter Lesson*), who advocate humility, spend a lot of time telling everyone else they're doing it wrong. Ultimately, both arguments ignore the essential interaction between data, computation, and the architecture. While both data and computation are necessary, they are insufficient and depend on the architecture taking advantage of the increased scale.

Residual connections depend specifically on layered representations, iterative refinement, and gradient-based optimization, which are characteristics of deep learning. By contrast, methods like Support Vector Machines (SVMs) are inherently shallow, relying on convex optimization to directly map inputs into high-dimensional feature spaces and determine separating hyperplanes. Because SVMs don't employ iterative gradient descent across multiple layers, they neither suffer from vanishing gradients nor benefit significantly from more data or compute, rendering innovations like residual connections irrelevant to their performance. If *The Bitter Lesson* tells researchers to scale; Ilya reminds researchers to ask: scale what?

Scaling was no longer theoretical and ResNet proved it. In 2012, AlexNet ignited the deep learning revolution, catapulting ImageNet Large Scale Visual Recognition Challenge (ILSVRC) into the defining battlefield for computer vision. Yet barely half a decade after and just two years after ResNet's introduction, ImageNet reached a saturation point. The contest, once framed as ahead of its time by Jitendra Malik, had been solved and researchers shifted their gaze toward new frontiers. In the aftermath, the ImageNet dataset transitioned from a cutting-edge benchmark into just another baseline, occupying a more symbolic position in computer vision research akin to MNIST, a dataset of handwritten digits that has long served as an introductory learning tool and basic reference point for beginners rather than a source of ongoing innovation.

Ultimately, ResNet (and the residual connections it introduced) hit the sweet spot between innovation and practicality. Within months of publication, ResNet was incorporated into prominent educational curricula, including Stanford's influential CS231n course, a resource featured on *Sutskever's List*. By 2025, the original ResNet paper had amassed over 100,000 academic citations and roughly 250,000 on Google Scholar, making it the most-cited scientific paper of the 21st century across *all* fields.^[45] Virtually every new computer vision paper referenced, compared against, or built upon ResNet, solidifying its role as the standard architecture in computer vision. Residual connections also transcended computer vision, fueling breakthroughs in domains ranging from AlphaGo's mastery of board games and AlphaFold's protein-structure predictions to advances in language modeling like GPT-2. By enabling deep networks, residual connections became essential to modern artificial intelligence across domains.

- [1] A Google search performed on June 14 2025 found more than 10,000 results for "add a residual connection."
- [2] <https://www.comet.com/site/blog/resnet-how-one-paper-changed-deep-learning-forever>
- [3] <https://blogs.microsoft.com/ai/microsoft-researchers-win-imagenet-computer-vision-challenge/#:~:text=In%20the%20ImageNet%20challenge%2C%20the,entrants%20by%20a%20large%20margin>
- [4] <https://agustinus.kristia.de/blog/residual-net/#:~:text=September%202015%2C%20at%20the%20ImageNet,rate%20of%20the%202014%20winner>
- [5] [https://blogs.microsoft.com/ai/microsoft-researchers-win-imagenet-computer-vision-challenge/#:~:text="We%20even%20didn't%20believe%20this,in%20Microsoft's%20Beijing%20research%20lab](https://blogs.microsoft.com/ai/microsoft-researchers-win-imagenet-computer-vision-challenge/#:~:text=)
- [6] [https://github.com/DeepLabCut/DeepLabCut/wiki/What-neural-network-should-I-use%3F-\(Trade-offs,-speed-performance,-and-considerations\)](https://github.com/DeepLabCut/DeepLabCut/wiki/What-neural-network-should-I-use%3F-(Trade-offs,-speed-performance,-and-considerations))
- [7] https://www.reddit.com/r/MachineLearning/comments/nbgb6a/d_are_resnets_as_good_as_it_gets/
- [8] https://ptidor.com/large_cnn.pdf
- [9] <https://arxiv.org/pdf/1611.05431>
- [10] <https://arxiv.org/abs/1409.4842>

- [11] <https://arxiv.org/abs/1602.07261>
- [12] https://ptidor.com/large_cnn.pdf
- [13] <https://arxiv.org/abs/1611.05431>
- [14] <https://arxiv.org/abs/1709.01507>
- [15] <https://www.frontiersin.org/journals/materials/articles/10.3389/fmats.2021.756798/full>
- [16] <https://pubmed.ncbi.nlm.nih.gov/37320940>
- [17] <https://neptune.ai/blog/self-driving-cars-with-convolutional-neural-networks-cnn#:~:text=In%20the%20case%20of%20Tesla%2C,50%20blocks>
- [18] <https://www.thinkautonomous.ai/blog/how-tesla-autopilot-works/>
- [19] <https://www.nature.com/articles/s41598-025-86743-z>
- [20] <https://arxiv.org/abs/1602.07261>
- [21] <https://arxiv.org/abs/2004.08955>
- [22] <https://arxiv.org/abs/1611.05431>
- [23] https://openaccess.thecvf.com/content_ICCV_2017/papers/He_Mask_R-CNN_ICCV_2017_paper.pdf
- [24] <https://arxiv.org/abs/1605.06431>
- [25] <https://arxiv.org/pdf/1511.07122>
- [26] <https://arxiv.org/abs/1411.4038>
- [27] <https://arxiv.org/abs/1412.7062>
- [28] <https://blogs.microsoft.com/ai/microsoft-researchers-win-imagenet-computer-vision-challenge/#:~:text=Microsoft%20researchers%20on%20Thursday%20announced,level%20performance>
- [29] <https://arxiv.org/pdf/1409.0575;>
- [30] Melanie Mitchell, Artificial Intelligence: A Guide for Thinking Humans
- [31] Hang Zhang, Chongruo Wu, Zhongyue Zhang, Yi Zhu, Haibin Lin, Zhi Zhang, Yue Sun, Tong He, Jonas Mueller, R. Manmatha, Mu Li, & Alexander Smola. (2020). ResNeSt: Split-Attention Networks. *See also* https://gluon-cv.mxnet.io/model_zoo/classification.html#id250 for code and visualization.
- [32] The top-1 accuracy is reported more frequently in recent years.
- [33] <https://www.wired.com/2015/01/karpathy/>
- [34] <https://arxiv.labs.arxiv.org/html/1409.0575>

- [35] <https://www.eetimes.com/microsoft-google-beat-humans-at-image-recognition/#:~:text=17,upped%20Microsoft%20by%200.04>
- [36] <https://www.popularmechanics.com/technology/a14116/microsoft-image-recognition-outperforms-humans/>
- [37] <https://www.theguardian.com/global/2015/may/13/baidu-minwa-supercomputer-better-than-humans-recognising-images#:~:text=Minwa%20is%20able%20to%20scan,respectively>
- [38] <https://www.entrepreneur.com/science-technology/can-artificial-intelligence-identify-pictures-better-than/283990#:~:text=dedicated%20human%20labeler%20who%20trained,1%20percent%20error%20rate>
- [39] https://ptidor.com/large_cnn.pdf
- [40] <https://www.wired.com/story/when-it-comes-to-gorillas-google-photos-remains-blind/>
- [41] <https://research.google/blog/train-your-own-image-classifier-with-inception-in-tensorflow/>
- [42] D. H. Wolpert and W. G. Macready, "No free lunch theorems for optimization," in IEEE Transactions on Evolutionary Computation, vol. 1, no. 1, pp. 67-82, April 1997,
- [43] <http://www.incompleteideas.net/Incldeas/BitterLesson.html>
- [44] <https://www.reuters.com/technology/artificial-intelligence/ilya-sutskever-how-ai-will-change-his-new-startup-safe-superintelligence-2024-09-05/>
- [45] <https://www.nature.com/articles/d41586-025-01125-9#:~:text=1>

4 Deep Learning Accelerates

This chapter covers

- The resurgence of recurrent neural networks (RNNs) after AlexNet
- Karpathy's blog made RNNs accessible, inspiring experimentation
- Chris Olah clarified LSTMs with vivid visuals and metaphors
- Selective dropout enabled deeper recurrent networks
- Deep Speech 2 proved the real-world potential of RNNs
- The engineering shift in artificial intelligence

Papers

- *The Unreasonable Effectiveness of Recurrent Neural Networks* (Karpathy, 2015)
- *Understanding LSTM Networks* (Olah, 2015)
- *Recurrent Neural Network Regularization* (Zaremba, Sutskever, and Vinyals, 2014)
- *Deep Speech 2: End-to-End Speech Recognition in English and Mandarin* (Amodei et al., 2015)

Recurrent neural networks (RNNs) are purpose-built for sequential tasks due to their ability to maintain and leverage information from previous inputs through hidden states. This allows RNNs to model temporal dependencies and handle inputs of varying lengths. Despite their theoretical promise, practical issues such as vanishing or exploding gradients, poor handling of long-range dependencies, and inefficient training often limit their effectiveness.

These challenges persisted until 2014, when Wojciech Zaremba, Ilya Sutskever, and Oriol Vinyals introduced specialized dropout methods explicitly designed for recurrent architectures. Their selective dropout strategy preserved crucial recurrent connections while regularizing the rest of the network, enabling deeper and more powerful RNNs that significantly improved language modeling, speech recognition, and machine translation performance. This innovation underscored Sutskever's conviction that advancements in AI rely less on inventing novel theories and more on disciplined, iterative engineering. Rather than chasing conceptual novelty, Ilya values rigorous experimentation and methodical refinement.

During this period, the intuitive appeal of recurrent networks gained widespread appreciation through influential educational resources. Andrej Karpathy's blog, "The Unreasonable Effectiveness of Recurrent Neural Networks," ignited excitement across the developer community with vivid examples and accessible code, sparking widespread experimentation. Complementing Karpathy's approachable insights, Chris Olah's influential essay, "Understanding LSTM Networks," provided clear, intuitive visualizations that illuminate the inner workings of RNNs.

The chapter concludes with an examination of Deep Speech 2 (DS2), a landmark end-to-end RNN-based speech recognition system that vividly illustrates how sophisticated architectures, combined with sufficient computational resources, can achieve near-human accuracy. DS2 represents precisely the practical, scalable progress that Ilya Sutskever advocates, cementing recurrent neural networks alongside convolutional neural networks as foundational architectures within the deep learning revolution.

4.1 The Unreasonable Effectiveness of Recurrent Neural Networks

In May 2015, Andrej Karpathy wrote a blog post titled "The Unreasonable Effectiveness of Recurrent Neural Networks." At that time, explanations of recurrent neural networks (RNNs) were dense, math-heavy, and difficult to follow. Karpathy offered something different: an intuitive approach demonstrating what RNNs could achieve through salient examples and easy-to-follow coding samples. He didn't just explain *how* RNNs worked; he showed them in action.

One particularly memorable experiment involved feeding the model the full text of Paul Graham's startup essays, who is best known for co-founding Y Combinator. The motivation was tongue-in-cheek: Graham offers timeless startup advice but is a slow generator. Wouldn't it be nice to synthesize fresh startup wisdom on demand?

Karpathy's "Paul Graham generator" was built as a character-level language model trained on approximately 1MB of concatenated text from Paul Graham's essays, totaling about one million characters. The model ("char-rnn") was implemented as a two-layer stacked LSTM, each layer comprising 512 hidden units, totaling approximately 3.5 million parameters. Both layers employed dropout regularization with a rate of 0.5. An LSTM is a widely used RNN variant equipped with specialized gates lacking in simple stacked RNN architectures. After training, the model generated new text by predicting one character at a time, sampling from a probability distribution whose randomness could be controlled by adjusting a temperature parameter; lower temperatures led to predictable but repetitive text, while higher temperatures produced more creative yet less coherent output. The resulting text is as follows:

LSTM (char-rnn)

"The surprised in investors weren't going to raise money. I'm not the company with the time there are all interesting quickly, don't have to get off the same programmers. There's a super-angel round fundraising, why do you can do. If you have a different physical investment are become in people who reduced in a startup with the way to argument the acquirer could see them just that you're also the founders will part of users' affords that and an alternation to the idea.[2] Don't work at first member to see the way kids will seem in advance of a bad successful startup. And if you have to act the big company too."

This... isn't great. While there are some familiar startup clichés, such as super angels, acquirers, programmers, and fundraising rounds, the vague shape of entrepreneurial advice is unmistakably lacking in coherence. It's the startup equivalent of the uncanny valley; vaguely familiar, but emotionally hollow. It even hallucinates a citation "[2]" which might be its only Paul Graham-esque aspect. Reading it now feels somewhat disorienting. Indeed, it fails to match the vibe test established by *Ovid's Unicorn*, a standard that wouldn't be reached by GPT-2 for another four years after this moment. It serves as a reminder of how low expectations once were, how easily many were impressed, and how far the field has come in subsequent years. Curious, I prompted ChatGPT (4o) to "help me rewrite this," and it responded coherently with the following:[1]

ChatGPT (4o)

This seems like a garbled or machine-generated paragraph, but here's an attempt to rewrite it into something coherent and meaningful, assuming it's about startup fundraising and acquisition dynamics.

Rewritten version: "Surprised investors weren't planning to raise money. I wasn't at the company full-time, but things were moving quickly—there wasn't a need to hire the same kind of programmers right away. It felt more like a super-angel round than a traditional fundraise. If your investment thesis is more physical than digital, you tend to back people who can reduce complexity in a startup. That makes it easier for potential acquirers to see the long-term value. Founders who focus on user needs early tend to build stronger momentum, even if the initial product doesn't work perfectly. And sometimes, if you're forced to act like a big company too soon, it can kill the very thing that made the startup interesting."

If we're being charitable, "char-rnn" is learning at the level of individual characters, making its mostly unintelligible rambling somewhat impressive. Character-level models struggle with coherence because they operate at a much finer granularity, generating text *one letter* at a time rather than using words or subwords. Unlike token-based models that work with meaningful chunks, character-level models have no built-in notion of words or syntax, making it harder to maintain semantic consistency across sentences. This means they must model long-range dependencies across many more steps, greatly increasing the risk of drifting off-topic or losing coherence. However, even operating at this granular level, the model isn't blindly assembling letters; it's approximating linguistic structure and conceptual patterns. Despite these inherent limitations, even barely coherent text generation felt groundbreaking in 2015.

Indeed, Karpathy's original post opens by capturing this sense of wonder: "There's something *magical* [sic] about Recurrent Neural Networks." At the time, this didn't sound as glib as it might today; watching a model generate text, character by character, was quite astonishing. Karpathy explains this "magic" by contrasting RNNs with convolutional neural networks (CNNs). CNNs require a fixed-size vector as input and produces a fixed-size vector as output, such as probabilities for different classes. This makes them ideal for independently analyzing static images or individual frames of video. However, this limits their effectiveness for tasks involving sequential data, such as sentences, audio streams, or continuous video sequences.

Structurally, RNNs share fundamental components with other artificial neural networks, including layers composed of neurons, connection weights, biases, activation functions, and training via backpropagation. Their distinguishing feature is their use of feedback loops, which allow them to handle sequences of varying lengths. Specifically, an RNN applies the same computational unit at each step, forming a chain-like structure that carries context forward.[\[2\]](#) See Figure 4.1.

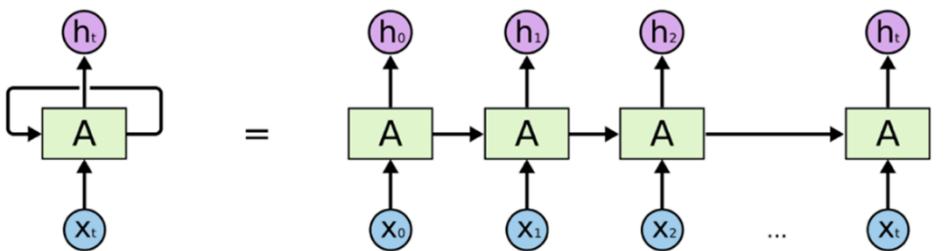


Figure 4.1 The image shows a chain-like structure representing a recurrent neural network (RNN). This structure illustrates how RNNs process data step-by-step, where the output from one step feeds into the next. The visual metaphor highlights that RNNs are designed explicitly for sequential data like time series, text, or audio. The repeating, linked modules reflect the RNN's ability to carry information forward, making them well-suited for learning patterns that unfold over time. *Olah (2015)*. Used with explicit permission granted by the author (Chris Olah).

At each timestep, a vanilla RNN takes two inputs: the current element of the sequence such as a word or character and the hidden state from the previous step, which encodes a summary of all prior inputs. This hidden state acts as a persistent internal “memory,” updating itself at every step and providing context for subsequent steps. While the network generates an output at each step, the hidden state, rather than the output, is passed along to the next step. Thus, the RNN maintains context over time, allowing it to model sequential data.

To better appreciate the significance of this design, consider a simpler model, such as an n-gram. N-grams store probabilities of sequences of length N . For example, predicting the next letter after “hel” relies solely on how often certain letters followed “hel” in the training data. They have a fixed, limited memory and cannot capture dependencies longer than N steps. By contrast, the RNN’s dynamic hidden state can encode contextual patterns beyond a fixed window. Moreover, instead of counting sequence frequencies, RNNs learn distributed representations that implicitly capture context, semantics, and linguistic structure, enabling them to model much more complex relationships.

When processing the word “hello,” the RNN doesn’t predict each letter in isolation; instead, it continuously updates an internal representation based on context. After encountering “hell,” the model must determine if the word is complete (“hell”) or if it should continue to form “hello.” Crucially, this decision isn’t merely based on memorized frequencies from training data but also on the immediate context encoded within its hidden state. In short, the hidden state lets the model “remember” and use context to resolve ambiguities like repeated letters and deciding when sequences should end.

Karpathy’s core demonstration focused on character-level prediction from datasets such as Wikipedia, Shakespeare, LaTeX, and Linux source code. The Wikipedia model produced a plausible markdown and citation structure. The Shakespeare model successfully imitated the structural patterns of stage dialogue, though not its dramatic impact or poetic quality. For example:

LSTM (char-rnn)

“PANDARUS:

*Alas, I think he shall be come approached and the day;
When little strain would be attain'd into being never fed.”*

The LaTeX model frequently generated syntactically valid equations but also introduced mistakes. For example, the model would begin with `\begin{proof}` environment only to mistakenly close it with `\end{lemma}`. These errors likely resulted from an inability to model long-term dependencies. That is, by the time the model reached the end of a proof, it had lost track of whether it was handling a proof or a lemma. Similarly, the model might open with `\begin{enumerate}` without remembering to close it. Although these errors became less frequent as the models grew larger and more sophisticated, such issues nonetheless highlight the challenge of maintaining context over extended sequences.

To further illuminate these results, Karpathy introduced tools for interpreting the internal workings of the RNN. He visualized neuron activations, revealing cells that fired in response to quotes, URLs, or markdown tags. No supervision or feature engineering was required; the model discovered these patterns purely from data. Some neurons became scope trackers, while others appeared to be quote balancers or indentation counters. These visualizations provided a rare window into the distributed computation within models, serving as an early and accessible example of the structure in learned representations.

The blog went viral, becoming the top story on Hacker News, with over 900 points and hundreds of comments.^[3] Readers praised Karpathy's ability to clarify complex concepts; one commenter noted it was "the first time I understood someone from this field... [he] just broken down the wall. What's more important, passion flows through his writing. And it can be felt. I got so excited while reading it."^[4] The sentiment was widely shared. On Reddit, readers marveled at the model's ability to generate plausible LaTeX formulas. "It can nearly write valid LaTeX?" someone wrote earnestly. "If that's not a sign of remarkable intelligence, I don't know what is." At least for Redditors, the true hallmark of "intelligence" is proper equation formatting. Valid or not, some vibes were there.^[5]^[6]

The blog's accompanying open-source code accumulated nearly 12,000 stars on GitHub and thousands of forks.^[7] Developers worldwide began training character-level RNNs on every dataset imaginable: cooking recipes, Eminem lyrics, Irish folk songs, Trump and Obama speeches, and even the Bible.^[8]^[9]^[10]^[11]^[12]^[13] Twitter and Reddit were filled with whimsical RNN-generated texts. These spin-offs created a flywheel effect: derivative content amplified the blog's reach, reinforcing its visibility and accelerating its influence across technical and popular domains.

AI researchers extended the post's reach. Chris Olah referenced Karpathy's blog in his viral essay, "Understanding LSTM Networks."^[14] Olah wrote, "I'll leave discussion of the amazing feats one can achieve with RNNs to Andrej Karpathy's excellent blog post." This was a pedagogical moment. Karpathy's clear explanations, examples, and code reduced the barrier to entry. One commenter captured the spirit: "Karpathy is one of my favorite authors... he spends much of his time demystifying the field... a brilliant way to encourage others to explore it."^[15] Karpathy's influential blog and the CS231n course became canonical, earning places on *Sutskever's List*, reflecting Ilya's reverence for Karpathy's work.

Naturally, not everyone was swept up by the vibes. A parallel conversation emerged, characterized by skepticism. Researcher Yoav Goldberg, for instance, wrote a follow-up blog comparing Karpathy's character-level RNNs to simpler baselines, such as unsmoothed n-gram models. Goldberg's critique echoed Alyosha Efros's earlier challenge to CNN researchers: just as Efros insisted on testing CNNs against basic nearest neighbor methods on identical data, Goldberg emphasized that character-level RNNs should be evaluated against straightforward statistical models. Both make the same fundamental point: simpler models perform surprisingly well given enough data. The point for skeptics is that data acts as an equalizer, neutralizing architectural differences. Yet, in reality, performance emerges not as a compromise between architecture and data, but rather from their combined strengths. The answer is not somewhere in the middle; it's emphatically both.

Goldberg argues that readers of Karpathy's blog were "impressed for the wrong reasons," a remark that suggests elitism. This argument implies that only a more discerning reader like Goldberg would understand what's *actually* impressive. The "wrong reasons," as Goldberg explains, are that readers mistakenly believe coherent text generation is unique or special to architectures such as RNNs which advanced models, like Transformers, would soon show the world. However, by emphasizing the arbitrary overlap of outcomes (i.e., coherent text) and dismissing architectural differences, Goldberg's critique creates a false equivalency. It's a shallow symmetry that collapses under inspection. These differences influence how these systems operate and how they can meaningfully utilize data.

A closer look reveals that Goldberg's critique was not just methodological; it had a rhetorical bite, primarily since Tomas Mikolov, best known for developing *word2vec*, provided clear quantitative evidence demonstrating the distinct advantage of RNNs over n-gram models.^[16] Mikolov's research in 2010 demonstrated measurable improvements with identical training datasets.^[17] RNNs achieved roughly 50% lower perplexity and an 18% relative reduction in word error rates compared to smoothed n-grams.

Even more telling, the RNN maintained its advantage even when the n-gram model was given vastly more data, highlighting a fundamental representational superiority. Thus, Goldberg's demonstration was deliberately provocative. Referencing an unsmoothed n-gram in this context was akin to declaring, "Your model isn't far from just a statistical copy-paste," or like a child on the playground asserting, "So what? I can do that, too." It's meant to cut something down to size by insisting the performance is just a trick anyone can do with enough data. The point isn't to engage; it's to dismiss. While RNNs had a long way to go, they demonstrated a qualitative and quantitative leap in learning structure over simple statistical models.

Comparing n-gram models with recurrent neural networks is misleading, as superficial similarities in outputs can be confused with architectural differences. Ironically, Goldberg's rhetorical trick of equating performance with architecture is commonly used by friends of artificial general intelligence, who conflate machine and human performance with architectural equivalence. This conflation exemplifies the core principle of computational functionalism, where biological and computational substrates are considered irrelevant as long as the functional role is maintained. From this perspective, similar performance, behaviors, or individuated functions render the specifics of their internal mechanisms insignificant. However, this reasoning flattens mechanistic and epistemic asymmetries between minds and models. The rhetorical trick in each scenario involves exaggerating similarities to obscure distinctions, disguising philosophical and practical differences behind superficial equivalence. The point here is that skeptics, like Goldberg, and AGI enthusiasts employ the same flawed reasoning. Regardless, Karpathy linked to Goldberg's critique in his original post, trusting readers to evaluate competing perspectives fairly.

Ultimately, Goldberg acknowledges that RNNs are "cool." However, even this concession feels less like admiration and more like a faintly dismissive nod to their popular appeal, as if they are entertaining toys rather than serious tools.^[18] At the time, this skepticism was justified, since RNNs were experimental, even toy-like, like n-grams. Yet, RNNs still demonstrated superior performance. Regardless, recurrence would soon be eclipsed by attention-based artificial neural networks.

Unlike Goldberg’s blog, Karpathy’s inspired countless projects precisely because it celebrated practical exploration and engineering ingenuity rather than dismissing them. It arrived precisely when deep learning was opaque to most developers, making it approachable, visual, playful, and free of pretense. It wasn’t merely about RNNs producing coherent output, but about sudden accessibility. Anyone with a GPU and a text file could explore it. Karpathy’s blog captured that brief cultural window when the gap between cutting-edge research and hobbyist experimentation momentarily collapsed, making artificial neural networks feel tangible to many for the first time. It demonstrates that a well-crafted blog can command mindshare rivaling that of a highly cited research paper. The truly “unreasonable effectiveness” was how far one well-written blog post could reach.

4.2 Understanding LSTM Networks

Chris Olah’s influential blog, “Understanding LSTM Networks,” set a new standard in clarity and accessibility for recurrent neural networks. Whereas Karpathy demonstrated *what* RNNs could achieve, Olah vividly illustrated *how* they achieved it, using intuitive visualizations and memorable analogies. His clear, engaging style didn’t merely simplify LSTMs; it set a new standard for accessibility in the field, later inspiring influential projects like the *Distill* journal, which Olah co-founded.[\[19\]](#) Although *Distill* ceased publication in 2021, its articles featured dynamic visualizations and interactive explanations precisely when the field most needed clarity, enabling readers to explore and intuitively grasp complex machine learning concepts.

Traditional feed-forward neural networks treat each input independently, but many tasks, like predicting the next word in a sentence, require context. RNNs solve this problem by incorporating a built-in memory, processing data step-by-step while maintaining a hidden state that captures past information. Unlike simple language models such as n-grams, which only look at a limited and fixed window (N), RNNs can capture dependencies spanning many steps, allowing earlier context to influence later predictions.[\[20\]](#)[\[21\]](#) Consider the sentence: “I grew up in France... I speak fluent French.” To accurately predict “French,” the network must recall the earlier mention of “France,” even if separated by many intervening words.

However, standard RNNs face practical limitations in modeling such “long-term dependencies.”[\[22\]](#) Training RNNs relies on backpropagation through time (BPTT), which “unrolls” the recurrent structure across multiple steps, effectively transforming the model for a loop-like recurrence into a deep linear sequence with shared weights. Predicting the next word after observing 100 words is analogous to training a convolutional network such as AlexNet with 100 layers. Keep in mind, AlexNet was just eight layers. As with CNNs, depth introduces challenges: gradients may either vanish, becoming too weak to learn distant dependencies, or explode, becoming excessively large, causing training instability.

In practice, vanishing gradients prevent RNNs from effectively capturing theoretically long-term dependencies across many time steps. For instance, learning the relationship between the first and last words of a 100-word sentence is extremely difficult because gradients encoding these dependencies become negligibly small. Put vividly, the influence of a piece of input on the loss diminishes exponentially as the gap between input and output increases, causing RNNs to ignore earlier, yet relevant information.[\[23\]](#)

To overcome the vanishing gradient problem and better model long-range dependencies, researchers developed a specialized type of RNN called Long Short-Term Memory networks (LSTMs).^[24] Olah famously described the internal workings of an LSTM as akin to a conveyor belt moving smoothly through a factory, selectively adding, carrying, and removing items (information) at precisely the right moments to illustrate how these networks effectively manage information over long sequences.^[25]

LSTMs manage information flow through specialized gates and maintain two types of memory: long-term and short-term—hence the name Long Short-Term Memory. The cell state, considered long-term memory, is the core idea behind LSTMs and allows information to pass through multiple steps almost unchanged, thereby preventing gradient issues. This is conceptually similar to residual connections in that both provide paths allowing gradients to flow more easily. However, while residual connections directly add input to output, LSTM cell states selectively manage information through gating mechanisms.

Using Olah’s metaphor, the cell state is like a conveyor belt and the backbone of the LSTM’s ability to retain important information much earlier in the sequence, enabling the network to capture long-term dependencies. In contrast, the short-term memory, or hidden state, is actively adjusted at each step and responds to immediate input. These two memory pathways are closely connected, allowing the LSTMs to manage long-term context and short-term variations.

To manage this dual (i.e., long-term and short-term) memory system, LSTMs use three interacting gates: the forget gate, the input gate, and the output gate.

The *forget gate* determines how much of the previously stored long-term memory to retain or discard. It combines the current input with the previous short-term memory and passes this information through a sigmoid activation function that produces values between 0 and 1. A value close to 0 means that most or all of the previous memory is forgotten, while a value near 1 indicates that memory is almost entirely preserved.

The *input gate* controls the introduction of new information into long-term memory. The network creates a candidate memory using the tanh activation function, generating values between -1 and 1. A separate sigmoid function determines which parts of this candidate memory should be stored. Technically, this is done with element-wise multiplication, meaning each element can be scaled individually rather than as a uniform process across all dimensions. The selected portion of candidate memory is added to long-term memory, thereby updating the cell state.

Lastly, the *output gate* shapes the short-term memory output. After updating the long-term memory, the cell state is processed through a tanh activation to produce a potential short-term memory. The output gate then uses another sigmoid function to decide what portion of this potential short-term memory to output as the actual short-term memory. This short-term memory also becomes the unit’s output for the current time step.

The simultaneous use of sigmoid and tanh activation functions is a particularly clever design choice. The sigmoid function acts as a gate modulating information flow by smoothly transitioning between fully closed (0) and fully open (1). Its natural output range (0 to 1) makes it ideal for selectively enabling or disabling parts of memory; effectively turning information “on,” “off,” or partially activated. This gating mechanism supports decision-like mechanisms such as:

- How much old information to discard (Forget Gate)
- How much new information to integrate (Input Gate)

- How much of the internal memory to expose at each step (Output Gate)

In contrast, the tanh function maps values between -1 and 1. This symmetry makes tanh ideal for creating candidate memory states, allowing the network to encode and integrate positive and negative updates. By capturing nuanced shifts in memory, tanh enables the internal states of the LSTM to move in multiple directions. Together, sigmoid and tanh complement each other:

- *Sigmoid gates* regulate information flow, determining what is retained, updated, or discarded.
- *Tanh layers* produce expressive memory capable of capturing complex, nuanced information rather than binary on/off decisions.

These gates and their carefully structured interactions ensure that information flows smoothly through the LSTM, without the unstable gradient fluctuations seen in vanilla RNNs. For example, when processing a story mentioning “Alice,” an LSTM can retain that context across multiple sentences, correctly predicting pronouns like “she” to refer back to Alice later. LSTMs selectively retain or discard context through these gating mechanisms, significantly enhancing performance in tasks that require long-range memory, such as language modeling, speech recognition, and handwriting recognition. This innovation has made LSTMs the standard RNN variant for sequence modeling tasks involving extensive context dependencies.

Chris Olah’s blog appears on *Sutskever’s List* not because it introduced groundbreaking algorithms or novel theoretical insights, but because of its clarity, execution, and influence at scale. Olah’s blog didn’t merely explain networks clearly; it reached critical mass, setting a standard of intuitive rigor that the entire AI community gravitated toward. Indeed, most entries on *Sutskever’s List* share this trait: their significance comes less from pure novelty than from their exceptional execution and demonstrated ability to scale.

4.3 Recurrent Neural Network Regularization

To understand why Karpathy’s and Olah’s blog posts had such an impact in 2015, it’s essential to revisit the challenges recurrent neural networks faced just a year earlier. In 2014, recurrent neural networks appeared to have hit a wall. Despite their theoretical promise, progress had stalled. Improvements were incremental at best, requiring painstaking tricks, brute-force ensembles, or hybrid combinations—or sometimes all three. Every attempt to scale up RNNs ran headlong into overfitting. As a result, surpassing key thresholds on standard sequential modeling benchmarks like the Penn Treebank (PTB) seemed agonizingly out of reach. One particularly daunting barrier was the perplexity of 100. Perplexity measures how “surprised” a language model is by new words; lower perplexity indicates the model makes better predictions. Yet, achieving a perplexity below 100 at the time felt like an insurmountable challenge.

In 2014, Zaremba, Sutskever, and Vinyals significantly transformed the landscape of recurrent neural networks by carefully engineering a selective dropout approach tailored explicitly to LSTM architectures.[\[26\]](#) Their technique solved a frustratingly persistent challenge of scaling recurrent architectures without severe overfitting, which had hindered meaningful progress. Echoing AlexNet’s earlier rigorous methodological approach and targeted regularization innovations, their work marked a decisive turning point for RNNs, which had been stuck.

Recall, AlexNet employed dropout, which randomly sets neuron outputs to zero with a 50% probability during each training iteration. This temporarily “turns off” random subsets of neurons, effectively creating a slightly different network at each training iteration. This randomness compels neurons to develop robust features instead of relying on specific neuron combinations. During inference (or evaluation), dropout is disabled, and neuron outputs are scaled down by 0.5 in the case of AlexNet, as a computationally efficient approximation of averaging predictions from all the hypothetical individual networks created during dropout training. In this way, dropout achieves effects similar to model ensembling, significantly reducing overfitting without explicitly training separate models. While dropout roughly doubles the number of training iterations, it improves generalization and mitigates the model’s tendency to overfit training examples.

Yet, training RNNs was notoriously frustrating, like CNNs were several years earlier. It was so difficult, in fact, that one of the era’s most influential papers, cited over 8,000 times, was titled *On the Difficulty of Training Recurrent Neural Networks*.[\[27\]](#) In the 2012 report, the authors hit the nail on the head, pinpointing precisely why recurrent networks struggled so much: severe training instabilities such as vanishing and exploding gradients crippled their ability to model long-term dependencies. Moreover, when researchers tried scaling up these networks to become deeper or larger, they inevitably encountered severe overfitting, limiting generalization and effectively blocking meaningful progress.

The authors of *On the Difficulty of Training Recurrent Neural Networks* offered partial solutions, such as gradient clipping, to tame exploding gradients. Yet, the vanishing gradient problem persisted, remaining a significant bottleneck to training deeper RNN architectures. In follow-up experiments, these same researchers explored deeper networks—testing architectures with three recurrent layers—but achieved only marginal perplexity improvements. Even these modest gains required elaborate tricks, such as shortcut connections and layer-by-layer pre-training reminiscent of pre-AlexNet CNN strategies.[\[28\]](#) They warned explicitly that training could become “even more problematic as models grow deeper,” implying diminishing returns unless breakthroughs emerged. Confronted with these limitations, they explicitly left open the critical question of why deeper recurrent networks were so hard to optimize, pointing directly toward dropout as a promising future avenue.

Traditional dropout techniques did not work well with RNNs. They disrupted the sensitive recurrent connections essential for maintaining memory capabilities, causing severe performance degradation and amplifying noise within recurrent dynamics.[\[29\]](#)[\[30\]](#) Moreover, existing regularization methods at the time had relatively small improvements for RNNs.[\[31\]](#)[\[32\]](#) As a result, RNNs remained shallow, with one or two recurrent layers and relatively small parameter counts. Even the most influential RNN of that period achieved its best performance of 114 perplexity with only a single hidden layer.[\[33\]](#) The boldest attempts at the time pushed to two-layer networks, achieving a modest improvement to 107 perplexity. Beyond two layers, researchers weren't merely encountering diminishing returns; they hit an impenetrable wall.[\[34\]](#)

Faced with this barrier, researchers turned to ensembles and hybrid approaches to achieve incremental gains. For example, state-of-the-art in 2012 combined three distinct approaches: a Kneser-Ney 5-gram language model that modeled short, frequent word sequences; a cache-based language model that boosted probabilities for recently occurring words; and a recurrent neural network (RNN) enriched by semantic topic vectors derived from Latent Dirichlet Allocation (LDA) that provided broader contextual understanding.[\[35\]](#) While clever, this hybrid strategy was far from the elegant promise of end-to-end learning popularized by AlexNet and more akin to a Rube Goldberg machine.

Each method failed to get perplexity below 100 on the PTB. Kneser-Ney alone yielded a perplexity around 141, the cache-enhanced variant around 125, and the RNN-LDA combination around 114. The combination of an RNN and LDA had a modest improvement of about 10 points over a cache-enhanced variant of Kneser-Ney. Yet, by carefully mixing their predictions and tuning each component's influence to balance their strengths, an ensemble brought perplexity down to approximately 92. Later, the same authors expanded this approach by including multiple variations of neural topic models alongside traditional ones, ultimately achieving a perplexity of about 72.9.

While these complex combinations were not practically scalable, they successfully reduced perplexity, underscoring a troubling reality: no single RNN architecture, not even the more robust LSTM, could break the 100 perplexity barrier on its own. In fact, before 2014, single-model LSTMs routinely struggled against benchmarks like the Penn Treebank (PTB). An early LSTM study from 2012 reported only about an 8% perplexity reduction compared to standard RNNs, highlighting their limited gains.[\[36\]](#) Zaremba, Sutskever, and Vinyals noted that even "relatively small LSTMs (in terms of number of parameters) can easily overfit the training set." Without effective regularization, adding more hidden units or additional layers would drive training perplexity down, but quickly cause validation perplexity to shoot up due to overfitting. This inability to scale meant single-model PTB perplexities rarely dipped below 120 in the pre-dropout era, and making further progress seemed out of reach.

This “glue-it-together” strategy reflected computer vision in the pre-AlexNet era, when nearly all high-performing vision systems struggled and lacked architectural details that could scale or support end-to-end learning. These pre-AlexNet solutions depended on meticulously engineered representations stitched together, including hand-crafted hierarchical feature pipelines using local descriptors to detect and describe image patches, histograms of quantized descriptors, and spatial pyramids to encode global representations that were then paired with shallow models like support vector machines. Likewise, early language models were mosaics of specialized components. Only by carefully integrating complementary pieces could computer vision researchers surpass performance plateaus, an approach that would soon be replaced by simpler, unified architectures trained end-to-end.

The 2014 breakthrough by Zaremba, Sutskever, and Vinyals changed the game. Their key insight was to apply selective dropout exclusively to the non-recurrent connections vertically between layers, never to the horizontal recurrent connections that preserve memory over time. This way, recurrent pathways preserve historical information without disruption, meaning dropout will enhance robustness without compromising stability. This selective strategy provides effective regularization, encouraging neurons to form stable representations while ensuring recurrent connections remain intact to retain and propagate historical information across timesteps. By introducing a selective dropout regularization for LSTMs, they slashed Penn Treebank perplexities dramatically. A single “large” LSTM with dropout (1,500 units, 2 layers) achieved 78.4 test perplexity, shattering previous records and outperforming the 10-model ensemble of smaller RNNs that managed a perplexity of 80. This was a dramatic leap.

The authors demonstrated the power of their specialized dropout technique through four diverse, real-world applications:

Language Modeling (Penn Tree Bank Dataset): The authors tested medium-sized LSTM networks with 650 units per layer and a 50% dropout rate and large networks featuring 1,500 units per layer and a 65% dropout rate. The results were significant: Dropout reduced the test perplexity from a baseline of 114.5 to just 78.4. Dropouts enabled larger, previously impractical LSTMs to train effectively, paralleling AlexNet’s breakthrough. Furthermore, combining ten large dropout-regularized models further reduced perplexity to 69.5, underscoring dropout’s scalability.

Speech Recognition (Icelandic Speech Dataset): Limited datasets often cause overfitting in acoustic modeling, leading to poor generalization. Dropout directly tackled this problem. While standard LSTM models reached 68.9% accuracy, dropout-regularized versions improved significantly to 70.5%. Although training accuracy decreased slightly due to dropout-induced noise, generalization improved substantially, replicating the core insight originally shown by AlexNet.

Machine Translation (English-to-French WMT Dataset): Regularized LSTMs demonstrated improved neural machine translation, boosting the BLEU score from 25.9 to 29.03. BLEU scores evaluate how closely machine-generated translations resemble human-written translations, where higher scores indicate better translation quality. Although traditional statistical methods remained competitive at the time, this improvement highlighted dropout’s clear potential for neural translation, laying a critical foundation for later breakthroughs such as the transformer architecture, which would soon revolutionize machine translation and NLP more broadly.

Image Caption Generation (MSCOCO Dataset): In the image captioning experiment, dropout improved the single-model BLEU score to 24.3, nearly identical to the 24.4 score obtained by an ensemble of multiple non-regularized models. This demonstrated dropout's efficiency advantage, as a single, well-regularized model achieved performance comparable to an ensemble, substantially reducing computational complexity.

Recurrent Neural Network Regularization is brief, with just six pages and fewer than 3,000 words, yet the paper demonstrated that the performance barrier was a solvable overfitting issue. In narrative terms, the field shifted from stagnation to breakthrough as perplexity scores that once barely changed suddenly dropped. By explicitly addressing regularization challenges, Zaremba, Sutskever, and Vinyals significantly expanded the practical applicability of deep learning, influencing subsequent research directions in speech recognition, machine translation, and sequence modeling.

4.4 Deep Speech 2

While convolutional neural networks had their defining moment with AlexNet's 2012 ImageNet victory, recurrent networks experienced a more incremental ascent, marked by steady advances across several fronts. LSTMs addressed the vanishing gradient problem of traditional RNNs, enabling better handling of long-term dependencies. However, their adoption remained limited due to computational constraints and persistent challenges with overfitting. A significant advance came in 2014 when Zaremba, Sutskever, and Vinyals introduced adaptive dropout specifically tailored for recurrent architectures.

Yet despite these methodological improvements, recurrent networks still awaited their "AlexNet moment." Several important papers nearly captured that definitive moment. The 2014 sequence-to-sequence paper by Sutskever, Vinyals, and Le showcased the compelling power of LSTMs in machine translation, sparking widespread interest.^[37] Later that same year, attention mechanisms were introduced in the influential paper *Neural Machine Translation by Jointly Learning to Align and Translate*, significantly improving translation quality and establishing new performance benchmarks.^[38] This paper is featured on *Sutskever's List* and will be analyzed in detail in Chapter 5. These advances were transformative but lacked the immediate and stark clarity of AlexNet's moment.

Baidu's Deep Speech 2 (DS2) provided the closest analogy to AlexNet for RNNs. DS2 convincingly demonstrated that large-scale, GPU-powered, end-to-end trained recurrent architectures could outperform traditional hybrid methods in speech recognition, achieving near-human-level accuracy in real-world conditions. With DS2, the potential of deep recurrent architectures finally crystallized: RNNs, particularly LSTMs, transitioned from being mere research tools to practical and commercially viable technologies.

Traditional speech recognition systems relied heavily on manually engineered pipelines: handcrafted acoustic features such as Mel-Frequency Cepstral Coefficients) and phonetic lexicons, requiring expertise, maintenance, and domain knowledge. Porting traditional pipelines to new languages or noisy environments was time-consuming and costly. Inspired by AlexNet's success in image classification, DS2 aimed to replicate the simplicity and efficacy of end-to-end learning for speech recognition. By mapping raw audio inputs (or minimally processed spectrograms) to textual outputs without intermediate hand-engineered components, DS2 sought to reduce complexity, enhance generalizability, and simplify deployment across languages and environments. However, realizing this vision posed significant computational and methodological challenges.

4.4.1 Core Architecture

Deep Speech 2 (DS2) employs an end-to-end architecture and is trained using a training objective known as Connectionist Temporal Classification (CTC). CTC is particularly important for speech recognition because spoken language is naturally variable in length, and CTC eliminates the need for precise audio-text alignment annotations. Alignment annotations refer to explicit labels indicating exactly which parts of an audio sequence correspond to particular characters, words, or phonemes in the transcript at each point in time. In traditional speech recognition pipelines, training typically required precise alignments. For instance, annotators would specify exactly where each phoneme or word starts and ends within the audio, resulting in detailed timing annotations. DS2 eliminates the need for these detailed alignment annotations through CTC, which treats the alignment between audio frames and output text as a latent variable, summing over all possible alignments during training. This allows the system to learn directly from pairs of audio and transcripts without explicit timing labels. Thus, DS2 greatly simplifies training and dataset preparation by removing the cumbersome requirement for manual alignment annotations. CTC-trained RNNs automatically segment, align, and filter unlabeled audio for high-quality training data.

Internally, DS2 uses a recurrent neural network (RNN), typically containing seven or more layers, along with convolutional layers at the front end. The authors explore architectures with up to 11 layers, including multiple recurrent and convolutional layers.

- *Convolutional Layers:* These input layers capture local patterns in the audio data, such as short-term variations in speech or slight frequency shifts caused by different speakers or microphones.
- *Recurrent Layers:* DS2 employed seven or more bidirectional recurrent layers, each using gating mechanisms to retain context over long sequences.
- *Fully Connected Output Layer:* This layer produces output probabilities for characters (or bigrams), directly translating audio into textual predictions.

Critically, DS2 introduced innovative techniques, including row convolution and striding with bigrams. *Row Convolution* allowed effective streaming speech recognition by enabling near-bidirectional performance from unidirectional RNNs, reducing latency for real-time use. This means a typical bidirectional RNN sees the entire input sequence (past and future contexts) to transcribe speech accurately. However, this is impractical in real-time streaming scenarios since future inputs are unavailable. To overcome this, row convolution uses a small “look-ahead” convolution that enables a unidirectional network to incorporate limited future context, simulating a near-bidirectional effect. Thus, row convolution balances the need for low-latency streaming speech recognition while achieving near-bidirectional performance.

Like row convolution, *striding* is a convolutional operation. It refers to periodically skipping small audio data segments, thus decreasing the number of input frames the RNN must process. Predicting bigrams rather than single letters ensures the output sequence better aligns with the compressed input representation, preserving accuracy.^[39] This is all pretty abstract. What is essential is that striding with bigrams was the more critical innovation of the two. Striding with bigrams reduces computational complexity and memory usage. It allowed the network to compress the temporal input data, dramatically decreasing the computational requirements while maintaining accuracy. Without this innovation, the scale and depth of DS2 would have been computationally prohibitive. Row Convolution, by contrast, was primarily a deployment optimization. It provided moderate improvements in accuracy for real-time, unidirectional streaming models. It was beneficial but more incremental.

Today, more advanced architectures have largely supplanted both row convolution and striding with bigrams. Attention-based transformers have replaced Row convolutions, while striding with bigrams has been replaced with subword tokenizers like byte-pair encoding. Yet, both were innovative at the time of DS2’s release.

4.4.2 Training Techniques

DS2 introduced several innovations to enhance computational efficiency. A key advancement was adopting multi-GPU synchronous stochastic gradient descent (SGD). Training involved distributing the workload across eight GPUs, each handling a subset of a minibatch, often totaling 512 samples. This strategy yielded near-linear scaling benefits, providing roughly a seven-fold increase in training speed compared to using a single GPU. Additionally, the authors optimized inter-GPU communication through a custom implementation known as an “all-reduce” operation. This method efficiently aggregated gradients computed independently on each GPU, dramatically improving communication speed and stability. The paper notes that these enhancements reduced training durations from weeks to a few days, enabling rapid experimentation. The models are adept at spelling without external language constraints and can implicitly disambiguate homophones. For example, “he expects the Japanese agent to sell it for two hundred seventy five thousand dollars.”

Another significant improvement was the shift to half-precision (FP16) arithmetic for inference. Using 16-bit floating-point operations reduced memory requirements by half and approximately doubled inference throughput on compatible GPUs, all without notable accuracy loss. The shift to FP16 by DS2 was an early and influential example of the broader concept now commonly referred to as *quantization*. Quantization generally involves representing network weights, activations, and computations with fewer bits, thereby reducing memory requirements and computation costs. FP16 arithmetic uses 16 bits (half-precision floating-point) instead of the conventional 32 bits (single-precision floating-point). By doing this, DS2 effectively halved the memory footprint and approximately doubled inference throughput on compatible hardware, demonstrating the potential gains of using lower-precision numerical formats without significantly compromising accuracy.

Quantization has become commonplace today, typically pushing even further down to 8-bit integers (INT8) and sometimes even lower-precision formats, especially for edge and mobile deployments. However, DS2's early adoption of FP16 showcased the viability and benefits of quantization techniques in practical, large-scale implementations, making it a milestone in developing model optimization strategies.

DS2's results can be primarily attributed to following the same principle established by AlexNet: significantly larger datasets lead to substantially improved performance. The authors trained DS2 on massive datasets. They used approximately 11,940 hours of English audio and around 9,400 hours of Mandarin audio, demonstrating that increasing data size yields near power-law reductions in Word Error Rate (WER). They discovered that each tenfold increase in training data volume resulted in roughly a 40% relative accuracy improvement, highlighting the importance of data scale. These results confirmed DS2's status as an RNN analog to AlexNet, demonstrating significant leaps forward by integrating end-to-end learning, large-scale training, and high-performance computational techniques.

The experimental results were groundbreaking, both in terms of model depth and accuracy. The authors demonstrated that deeper network architectures, typically comprising seven or more layers, consistently outperformed shallower models. Compared to simpler recurrent neural networks, these deeper structures, equipped with gating mechanisms such as LSTMs or GRUs, significantly enhanced transcription accuracy. As an aside, gated recurrent units, or GRUs for short, streamline the LSTM architecture, combining the cell and hidden states into one and reducing gating complexity to just two components: the update gate and reset gate.^[40] GRUs retain most of LSTM's advantages, but their simpler design is computationally lighter and easier to train. In practice, these innovations were often used together.

DS2's performance on standard benchmarks was particularly notable, achieving better-than-human word error rates on clean speech datasets such as WSJ and LibriSpeech. Moreover, it approached human parity even in highly challenging scenarios, including noisy environments and heavily accented speech, as demonstrated by results on the CHiME dataset. Additionally, the Mandarin version of the system achieved near-human accuracy on short voice queries, effectively bypassing the complexities associated with traditional, language-specific pipelines. These results underscored DS2's capacity to generalize effectively across diverse real-world speech contexts.

4.4.3 Architectural Enhancements

DS2 introduced key architectural enhancements that significantly improved the stability and efficiency of training deep recurrent neural networks. One crucial innovation was *sequence-wise batch normalization*, a variant of batch normalization explicitly adapted for recurrent models. Standard batch normalization (introduced in 2015) normalizes layer inputs during training by computing the mean and variance of activations within each mini-batch, ensuring consistent activation distributions.[\[41\]](#) This consistency reduces the phenomenon known as *internal covariate shift*, where the distribution of inputs to a layer continuously changes throughout training. By stabilizing input distributions at each layer, batch normalization enables higher learning rates, reduces sensitivity to weight initialization, and accelerates convergence. DS2's sequence-wise adaptation applied this concept specifically along the temporal dimension, stabilizing recurrent network training and enhancing generalization capabilities.

In the case of DS2, the authors modified batch normalization into *sequence-wise batch normalization*, explicitly adapting it for recurrent neural networks by normalizing activations across entire sequence lengths. Normalizing activations across each training mini-batch at every recurrent layer stabilized internal network dynamics, allowing for the training of significantly deeper recurrent architectures without numerical instability or vanishing gradients.

Another improvement was the introduction of a curriculum learning strategy called *SortaGrad*. With SortaGrad, training sequences were initially sorted by length, enabling the network to progressively tackle more challenging examples over time. By presenting shorter (and typically easier) utterances first, SortaGrad ensured more stable and efficient training in the early stages, effectively reducing numerical instability and helping the model converge reliably from the outset. Sequence-wise batch normalization and SortaGrad enabled DS2 to train deeper networks and achieve substantially improved transcription accuracy and training stability. DS2 incorporates synthetic background noise in approximately 40% of training examples, enhancing robustness to real-world noisy environments.

4.4.4 Language Models and Decoding

A series of carefully engineered optimizations aimed at real-time inference and computational efficiency made deploying DS2 as an interactive cloud service at scale practical. A key innovation was the introduction of "Batch Dispatch," a technique that dynamically grouped concurrent user audio streams into optimized batches. Even small batches of size 2–4 significantly reduced memory bandwidth overhead by reusing loaded weights. By intelligently batching inputs, DS2 significantly reduced the memory bandwidth overhead associated with processing individual requests separately, enabling extremely fast inference, often achieving latencies as low as 50 to 70 milliseconds, even under moderate system load.

Lastly, the team applied heuristic pruning methods during the beam search decoding process. Decoding schemas are widely used in sequence prediction tasks, such as machine translation, speech recognition, and language modeling. They provide an efficient way to find high-quality output sequences from probabilistic models without evaluating all possible sequences, which is often computationally infeasible. Beam search decoding maintains and progressively expands multiple high-probability candidate sequences at each prediction step, rather than selecting only the most probable token. Upon completion, it deterministically chooses the sequence with the highest cumulative probability, balancing computational efficiency with prediction quality. The heuristic pruning used by DS2 discards the least promising candidates based on their probabilities. This dramatically reduces the computational cost of decoding, making it much faster without substantially decreasing accuracy. This simplifies the computational complexity, which is particularly notable in Mandarin speech recognition scenarios with extensive character sets. The authors use a beam size of 500 for the English decoder and 200 for the Mandarin decoder.

4.4.5 Significance and Broader Impact

Deep Speech 2 (DS2) represented a breakthrough in automatic speech recognition, replacing complex, manually engineered pipelines with an end-to-end deep learning approach. It achieved impressive accuracy comparable to human performance in benchmark tests. However, DS2's claim of surpassing human-level accuracy was controversial. Critics noted that these benchmarks relied on Amazon Mechanical Turk workers, who, while proficient, were not professional stenographers or expert transcribers. This raised questions about DS2's true parity with human experts in realistic transcription settings. Nevertheless, DS2 demonstrated remarkable versatility, performing effectively in vastly different languages such as English and Mandarin with minimal modifications. This marked an important step toward a universal automatic speech recognition system capable of generalizing across diverse languages and challenging acoustic environments.

DS2 accelerated training by up to seven times, turning experiments that previously took weeks into days. Iterating rapidly over enormous datasets—nearly 12,000 hours of English and 9,400 hours of Mandarin speech—substantially boosted accuracy. Notably, DS2 exhibited clear scaling laws, achieving a 40% relative reduction in word error rate (WER) with each tenfold increase in data.

The observation that “WER decreases by 40% relative for each factor of 10 increase in training set size” is an early and notable example of empirical scaling laws in deep learning research. While the formal term “scaling laws” as a research concept wouldn't gain widespread popularity until 2020 with the publication of the “Scaling Laws for Neural Language Models,” the DS2 paper demonstrated an empirical relationship between dataset size and model performance. In other words, DS2 prefigured that deep learning performance improvements follow predictable, quantitative relationships tied to scale.

The architecture involved recurrent networks with seven or more bidirectional LSTM and GRU layers, significantly outperforming shallower models. Innovations such as sequence-wise Batch Normalization, “SortaGrad” curriculum learning, and convolutional front-ends with long stride inputs further stabilized training and reduced computation per example by threefold. DS2 also incorporated FP16 inference, doubling throughput, and employed beam search pruning heuristics to significantly accelerate decoding speeds, which is particularly beneficial for languages with extensive character sets like Mandarin. Practical deployment strategies, including optimized GPU kernels and the “Batch Dispatch” system, enabled real-time transcription at median latencies below 70 ms. This underscores how effectively DS2 married algorithmic innovation with raw computational power.

4.4.6 An Engineering Shift

In the 2010s, AI research shifted to large-scale, engineering-intensive systems emphasizing end-to-end learning. Deep Speech 2 (DS2) exemplified this direction, characterized by meticulous kitchen-sink engineering and reliance on massive datasets and extensive computational resources. Like AlexNet, these projects combined empirical strategies to achieve substantial real-world performance gains instead of singular theoretical breakthroughs. AlexNet integrated known techniques such as GPU optimizations, ReLU activations, dropout, and data augmentation to yield unprecedented results. Similarly, DS2 followed this pragmatic pattern and embraced end-to-end training while combining architectural refinements, computational optimizations, curriculum learning, and advanced normalization to improve performance significantly.

Despite impressive performance improvements, the engineering-focused approach of systems like AlexNet faced skepticism from the broader scientific community. Critics dismissed such advances as mere “engineering” or achievements driven by “scale,” implying these efforts represented a retreat from—or even a contradiction of—the pursuit of “true artificial intelligence,” a phrase whose oxymoronic nature hints at deeper tensions about authenticity and artificiality. This confusion is analogous to asking whether the number 3 has a color and conflating fundamentally different categories.[\[42\]](#)

For instance, one retrospective critique observed that AlexNet was “largely an evolutionary improvement over LeNet,” not a fundamentally new design. Yet this assessment overlooks how AlexNet, *not* LeNet, reshaped the entire field of computer vision.[\[43\]](#) Likewise, Gary Marcus cautioned that while AlexNet represented “important work with immediate practical applications,” it was ultimately “only a small step toward the creation of truly intelligent machines.”[\[44\]](#) This phrase illustrates how critics often measure progress against an idealized future. While we generally desire solutions ahead of their time, that notion is almost always incoherent. Incoherent because the future is not a source of knowledge that can ever be validated. Consequently, we cannot measure progress against the future, only against where we start solving a problem.

Genuine innovation emerges incrementally through real-world trial and error, highlighting the difference between those with “dirty hands,” constrained by the real-world limitations of knowledge, and those with “clean hands,” who critique progress against abstract, often unreachable standards. Marcus is correct; AlexNet’s achievement was impressive empirically, which Marcus invokes almost dismissively, as though pragmatism is a lesser form of innovation. Yet, progress was precisely what the field had been lacking. Ilya showed how to actually use the technology when the field was stuck. The transition from theoretical promise to tangible capability required precisely the empirical pragmatism that critics undervalued.

Likewise, DS2 was celebrated as a breakthrough. *MIT Technology Review* named Baidu’s system one of 2016’s top breakthroughs, noting it outperformed professional humans on some tasks.[\[45\]](#) Within the speech research community, there was excitement that a simple architecture could replace entire handcrafted pipelines and still outperform state-of-the-art systems. However, academic skeptics viewed Deep Speech 2 as an engineering tour de force rather than a source of insight.

Reviewers critiqued the work for a lack of novelty in the peer review for the International Conference on Machine Learning in 2016, where the DS2 paper was published. “My main issue with the paper is that a lot of it draws on existing work... and provides very little in terms of novelty,” one reviewer wrote, noting that “most of the techniques have all been tried before.” The review continues, “Right now, the paper just reads as sticking together a bunch of previous techniques.” Another reviewer echoed that DS2 introduced no fundamentally new methods, remarking: “The novelty of the paper is through scaling up and implementation rather than any methodological advance.”[\[46\]](#) In short, academic experts felt DS2 combined known ideas but didn’t reveal new algorithms or theoretical insights.

This engineering-focused approach contrasts with theoretical research exemplified by foundational papers, such as Martens’ Hessian-free optimization, Rumelhart’s backpropagation, or Hochreiter and Schmidhuber’s original LSTM. These contributions prioritize conceptual clarity, mathematical elegance, and novelty. DS2 and AlexNet, in contrast, emphasize solving tangible, large-scale engineering challenges with measurable impacts. One could almost be forgiven for mistaking the goal of AI research as making something useful. Ultimately, the tension here is unmistakable: theoretical novelty versus practical efficacy. A preference for one approach over another reveals fundamental differences in the researcher’s values and intellectual orientation.

AlphaGo’s historic victory over Go champion Lee Sedol in 2016 demonstrated the enormous potential of intensive engineering efforts. Before AlphaGo, most AI advances emerged from small, idea-driven teams or individuals: Martens’ Hessian-free optimization had a single author; Rumelhart’s backpropagation had three; Hochreiter LSTM paper involved two. While AlexNet included just three authors, Deep Speech 2 included thirty-four, and AlphaGo involved twenty. The field was maturing, becoming increasingly collaborative, resource-intensive, and focused on large-scale engineering.

For Ilya Sutskever, AlphaGo crystallized a conviction that had been developing for some time: large-scale, engineering, not merely novel theoretical ideas, would drive the future of AI. This was non-obvious in the mid-2010s. For example, at Google, Sutskever had encountered a culture deeply aligned with academia's preference for conceptual innovation. While valuable for generating ideas, Ilya recognized that conceptual breakthroughs alone would no longer suffice. Increasingly complex models demanded massive datasets, advanced GPU infrastructure, and robust engineering practices beyond the capabilities of small, academically oriented teams. Such groups were inherently resource-constrained, optimized for novelty rather than sustained, scalable, and reliable implementation.[\[47\]](#)[\[48\]](#)

Google was not the problem; academia itself presented deeper structural issues. According to Sutskever, academic incentives encouraged researchers to "think small," rewarding incremental advances and superficial novelty rather than practical, scalable impact. Recognizing this fundamental misalignment, OpenAI was designed to prioritize practical effectiveness and deployability over incremental novelty and academic accolades.

This philosophical divergence illuminates the motivation behind *Sutskever's List*, which implicitly highlights that modern AI is rooted in a small set of seminal papers, with most other publications introducing negligible or even negative value by introducing superficial variations that create noise rather than progress. The epidemiology of the problem is blurry, but M.E.J. Newman finds a strong first-mover effect in which the first academic papers in a field will receive citations at a rate enormously higher than papers published later, attracting researchers to what is seen as the future like moths to a light.[\[49\]](#)[\[50\]](#)

Ian Goodfellow's Generative Adversarial Networks (GANs) illustrate academia's incrementalism problem. After Goodfellow's groundbreaking 2014 paper, researchers produced over five hundred variants of GANs, yet minimal improvement resulted.[\[51\]](#)[\[52\]](#) It all begs the question: What is the practical value of the 501st GAN?[\[53\]](#) Where does it end? *Sutskever's List* offers an answer: prioritize depth and practical efficacy over accumulating trivial incremental knowledge.

Sutskever's skepticism toward novelty shapes his pragmatic research approach. Asked about the balance between generating entirely new ideas and refining existing ones, he stressed that conceptual breakthroughs represent only a modest fraction of genuine advancement. Real progress comes from understanding existing ideas deeply, iteratively improving and deploying them, and carefully analyzing results, particularly when experiments yield surprising outcomes.[\[54\]](#) In Ilya's eyes, deployment is a research tool that provides unattainable insights through purely theoretical or small-scale studies. This pragmatic philosophy mirrors John Carmack's vision for Keen Technologies, which deliberately revisits older ideas to uncover opportunities enabled by modern scale or an engineering-driven approach. Like Sutskever, Carmack values systematic refinement, scaling, and real-world validation of existing concepts.

As OpenAI's Chief Scientist, Sutskever steered the lab toward product-oriented research, favoring large-scale experiments over incremental papers. The GPT series exemplifies this philosophy. While Google introduced the Transformer architecture in 2017, OpenAI scaled it into practical, deployable products like GPT-3, GPT-4, and ultimately ChatGPT. For Sutskever, deploying systems used by millions offers greater value than novel ideas confined to academic circles. In his view, genuine progress arises from understanding and improving technologies rather than perpetually chasing novel concepts that dozens of people will read. This philosophy permeated OpenAI's culture and differentiates it from pure research organizations. At the same time that OpenAI was founded in late 2015, Google reintroduced itself as an AI-first company. While OpenAI was established with the goal of AGI, it became an engineering-first company under Ilya's leadership, which may have made all the difference.

Historically, ambitious AGI-oriented companies failed to balance aspirations with practical deployment. For instance, years before Ben Goertzel edited the book, "Artificial General Intelligence," his company, Intelligenesis, developed Webmind, a product envisioned as a disembodied "global brain" nurtured on the Internet. The media declared Webmind as "dumb," but with a "brilliant future."^[55]^[56] Publications like *The Wall Street Journal* and *The Australian Financial Review* envisioned Webmind ushering in a new era of intelligence that would eclipse humanity's cognitive supremacy.^[57] All the while, Intelligenesis was going bankrupt because of its solve-no-problem business model. Goertzel wrote in a business postmortem that the goal of "creating a thinking machine and then commercializing it" should have been "laughed out of any conversation with any serious businessperson."^[58]^[59] Webmind stands in stark contrast to OpenAI, which has favored an incremental, iterative approach, deploying practical, problem-solving systems such as GPT and ChatGPT, each carefully aligned with user needs and market viability.

Sutskever has *relentlessly* focused on experimentation, even when the underlying algorithms, such as CNNs, RNNs, or Transformers, were already well-established. Ilya did not invent these architectures; he made them work. When asked how much time he devotes to entirely new ideas versus refining existing ones, he replied decisively: "Coming up with entirely new ideas is a modest part of the work." Far more crucial, he explained, is "to understand the results, the existing ideas, and what's really happening—especially when experiments yield unexpected outcomes." Ilya's genius is taking ideas that looked great on paper and making them work in the real world.

- [1] <https://chatgpt.com/share/67fe5d24-1cec-8002-83e2-969c3bb400a5>
- [2] Olah, C. (2015, August 27). Understanding LSTM networks. colah's blog. Retrieved from <https://colah.github.io/posts/2015-08-Understanding-LSTMs/>
- [3] <https://news.ycombinator.com/front?day=2015-05-21#:~:text=The%20Unreasonable%20Effectiveness%20of%20Recurrent,18%20>
- [4] <https://news.ycombinator.com/item?id=9584325#:~:text=,others%20to%20explore%20it>
- [5] https://www.reddit.com/r/programming/comments/36su8d/the_unreasonable_effectiveness_of_recurrent/
- [6] <https://cvoss.com/modeling-molecules-with-rnns>
- [7] <https://github.com/karpathy/char-rnn>

- [8] <https://gist.github.com/nylki/1efbaa36635956d35bcc>
- [9] https://www.reddit.com/r/MachineLearning/comments/36zi75/i_used_andrej_karpathys_charrnn_to_compose_irish/
- [10] <https://github.com/ryanmcdermott/trumpgen>
- [11] <https://x.com/MrChrisJohnson/status/605783907468865536>
- [12] https://x.com/RNN_Bible
- [13] <https://medium.com/@samim/obama-rnn-machine-generated-political-speeches-c8abd18a2ea0>
- [14] Olah, C. (2015, August 27). Understanding LSTM networks. colah's blog. Retrieved from <https://colah.github.io/posts/2015-08-Understanding-LSTMs>
- [15] <https://news.ycombinator.com/item?id=9584325>
- [16] <https://arxiv.org/abs/1301.3781>
- [17] https://www.isca-archive.org/interspeech_2010/mikolov10_interspeech.html
- [18] <https://nbviewer.org/gist/yoavg/d76121dfde2618422139>
- [19] <https://distill.pub>
- [20] Yoshua Bengio, Rejean Ducharme and Pascal Vincent. 2003. A neural probabilistic language model. *Journal of Machine Learning Research*, 3:1137-1155
- [21] Convolutional networks can be used to approach problems where time and sequences are a consideration; however, such architectures are cumbersome. <https://arxiv.org/pdf/1603.06995.pdf>
- [22] Y. Bengio, P. Simard and P. Frasconi, "Learning long-term dependencies with gradient descent is difficult," in IEEE Transactions on Neural Networks, vol. 5, no. 2, pp. 157-166, March 1994, doi: 10.1109/72.279181
- [23] Hochreiter, Sepp; et al. (15 January 2001). "Gradient flow in recurrent nets: the difficulty of learning long-term dependencies". In Kolen, John F.; Kremer, Stefan C. (eds.). *A Field Guide to Dynamical Recurrent Networks*. John Wiley & Sons. ISBN 978-0-7803-5369-5.
- [24] Sepp Hochreiter; Jürgen Schmidhuber (1997). "Long short-term memory". *Neural Computation*. 9 (8): 1735–1780. doi:10.1162/neco.1997.9.8.1735. PMID 9377276. S2CID 1915014.
- [25] Olah, C. (2015, August 27). Understanding LSTM networks. colah's blog. Retrieved from <https://colah.github.io/posts/2015-08-Understanding-LSTMs>
- [26] <https://arxiv.org/abs/1409.2329>
- [27] <https://arxiv.org/abs/1211.5063>

- [28] <https://arxiv.org/pdf/1312.6026>
- [29] Formally introduced by Srivastava, Hinton, Krizhevsky, Sutskever, and Salakhutdinov in 2013 (after AlexNet): <https://www.cs.toronto.edu/~hinton/absps/JMLRdropout.pdf>
- [30] <https://arxiv.org/abs/1311.0701>
- [31] Srivastava, Nitish. Improving neural networks with dropout. PhD thesis, University of Toronto, 2013.
- [32] Graves, Alex. Generating sequences with recurrent neural networks. arXiv preprint arXiv:1308.0850, 2013.
- [33] https://www.microsoft.com/en-us/research/wp-content/uploads/2012/07/rnn_ctxt_TR.sav.pdf
- [34] <https://arxiv.org/pdf/1312.6026>
- [35] https://www.microsoft.com/en-us/research/wp-content/uploads/2012/07/rnn_ctxt_TR.sav.pdf
- [36] <https://www-i6.informatik.rwth-aachen.de/publications/download/820/Sundermeyer-2012.pdf>
- [37] <https://arxiv.org/pdf/1409.3215>
- [38] <https://arxiv.org/abs/1409.0473>
- [39] Using bigrams—pairs of characters—instead of single characters (unigrams) helps match the temporal reduction introduced by aggressive striding.
- [40] <https://arxiv.org/abs/1412.3555>
- [41] <https://arxiv.org/abs/1502.03167>
- [42] Echoing Wittgenstein’s critique in The Blue and Brown Books, describing computation in human terms is akin to asking if “the number 3 has a colour,” thus conflating fundamentally different categories.
- [43] http://d2l.ai/chapter_convolutional-modern/alexnet.html
- [44] <https://www.newyorker.com/news/news-desk/is-deep-learning-a-revolution-in-artificial-intelligence#:~:text=York%20Times%20suggests%3F%20There%20is,page>
- [45] <https://www.technologyreview.com/technology/conversational-interfaces/>
- [46] <https://icml.cc/2016/reviews/76.txt#:~:text=Justification%3A%20This%20paper%20is%20very,for>
- [47] <https://www.dwarkesh.com/p/ilya-sutskever>
- [48] https://youtu.be/H1YoNlz2LxA?si=03KgI1XB_LxVb0V7&t=595

- [49] M. E. J. Newman: "The First-Mover Advantage in Scientific Publication," *Europhysics Letters* 86, no. 6 (June 2009), <https://iopscience.iop.org/article/10.1209/0295-5075/86/68001>.
- [50] Ryan Hill and Carolyn Stein find evidence to support that getting scooped will damage the likelihood of publication and the likelihood of appearing in a top journal, as well as the likelihood of fewer citations, though these likelihoods are much more moderate than scientists actually believe.
- [51] A. Hindupur, "The GAN Zoo," GitHub, <https://github.com/hindupuravinash/the-gan-zoo>.
- [52] Lucic, M., Kurach, K., Michalski, M., Gelly, S., & Bousquet, O. (2018). Are GANs Created Equal? A Large-Scale Study. In Advances in Neural Information Processing Systems. Curran Associates, Inc.
- [53] Sridhar Mahadevan, "Does Every Paper in Machine Learning Introduce a New Algorithm?" Quora, August 31, 2020, www.quora.com/Does-every-paper-in-machine-learning-introduce-a-new-algorithm/answer/Sridhar-Mahadevan-6?ch=10&share=4657945d&srq=1Rf.
- [54] <https://www.dwarkesh.com/p/ilya-sutskever>
- [55] Petzinger Jr., Thomas. "Mathematician Perceives Mind as a Company-Intranet Model." The Wall Street Journal. May 22, 1998. <https://www.wsj.com/articles/SB895791428926727000>.
- [56] Ben Goertzel, "Waking Up from the Economy of Dreams," April 9, 2001, goertzel.org/benzine/WakingUpFromTheEconomyOfDreams.htm.
- [57] Cave, M. (2000, May 11). One dumb baby with a brilliant future. Australian Financial Review. <https://www.afr.com/politics/one-dumb-baby-with-a-brilliant-future-20000511-k9g4u>.
- [58] Goertzel's new company, TrueAGI, will focus on the enterprise to offer the "mind-as-a-service."
- [59] S. Shead, "Alphabet's DeepMind Losses Soared to \$570 Million in 2018," Forbes, August 7, 2019. Thomson, Amy. "Google Waives \$1.5 Billion DeepMind Loan as AI Costs Mount." Bloomberg.com, Bloomberg, Dec. 17, 2020.