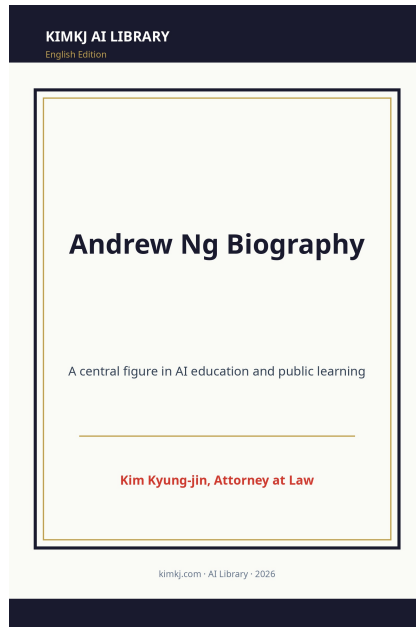




Andrew Ng Biography

A central figure in AI education and public learning

Kim Kyung-jin, Attorney at Law

AI Library - kimkj.com - 2026

Table of Contents

1. From London to Stanford
2. An Early Researcher: Robots and Reinforcement Learning
3. Google Brain and the Scaling Experiment
4. Three Years at Baidu
5. Coursera and Lowering the Threshold for Online Education
6. DeepLearning.AI: AI for Everyone
7. Landing AI and Artificial Intelligence on the Enterprise Floor
8. AI Fund, Amazon, and the Startup Ecosystem
9. Realism in the Face of AGI Fever

From London to Stanford

On April 18, 1976, a child was born in London, England. Given the name Andrew Yan-Tak Ng, this child's parents were immigrants from Hong Kong. Because of this family background, he came to hold both a Hong Kong British identity and, later, a Hong Kong American one. The surviving records do not reveal exactly what his parents did for work or what circumstances led them to settle in Britain. No record survives that would let us guess at his childhood personality or inner conflicts. This chapter does not fill those gaps with imagination. Following only confirmed facts, it traces the path of a person born in London who passed through Asia and American universities on his way to Stanford.

Young Ng spent much of his formative years in Hong Kong and Singapore. One can guess that moving between the two cities helped cultivate an international sensibility, but no specific anecdote survives to support this. Nor does any document record when he first encountered a computer or what drew him to mathematics and science. One clear milestone does exist, however. In 1992, he graduated from Raffles Institution, a prestigious school in Singapore. Raffles Institution has a long history and reputation in Singapore, and many of its graduates have gone on to make names for themselves in various fields. Ng joined that lineage, but no record survives of which classes influenced his path during his time there or which teachers or friends he encountered. This period, then, remains only a single line of fact. Born in London in 1976. Raised in Hong Kong and Singapore. Graduated from Raffles Institution in 1992. These three sentences are all that can be confirmed about his life before he turned twenty.

After leaving Singapore, Ng headed to Carnegie Mellon University in Pennsylvania, USA. Carnegie Mellon was a world-class school in computer science, one that would come to be called a cradle among artificial intelligence researchers for the talent it produced in the field. There, Ng became a triple major, simultaneously completing degrees in computer science, statistics, and economics. Each of the three disciplines demands a different way of thinking. Computer science trains a person to break problems down into algorithms, statistics builds an eye for reading patterns in uncertain data, and economics teaches one to examine how choices are made under limited resources. Absorbing all three at once could not have been an easy schedule, yet Ng completed the program and graduated first in his class in 1997 with his bachelor's degree. No recorded recollection from him explains why he chose this particular combination of majors or which moment during his undergraduate years shaped his academic direction. Judging by the outcome, though, this combination built the foundation he would need to grow into a researcher who could handle mathematical rigor, statistical intuition, and practical judgment together in the field of machine learning.

Overlapping with his undergraduate years, from 1996 to 1998, Ng had the opportunity to work as a researcher at AT&T Bell Labs. Bell Labs was a place that produced countless

innovations in telecommunications and computer science in the latter half of the twentieth century, including the invention of the transistor and research that laid the foundations of information theory. At that laboratory, the subjects Ng worked on were reinforcement learning, model selection, and feature selection. Among the various methodologies of machine learning, reinforcement learning is an approach in which, instead of a person supplying the correct answer for every case, a system learns to find better actions through trial and error. It is similar in principle to training a dog by rewarding desired behavior and withholding reward otherwise, refined into a mathematically precise form. Model selection is the task of determining which of several candidate models best explains given data, while feature selection is the task of picking out the few variables that actually matter from among many. The very fact that a young man barely past twenty handled these three subjects foreshadowed the direction his later career would take. No firsthand recollection from him confirms whether this period was the decisive turning point that led him onto the path of a full-fledged artificial intelligence researcher. But since the subjects he worked on during this time reappear as the core concepts running through his entire doctoral dissertation, his years at Bell Labs can be seen as the place where his academic roots were planted.

After graduating from Carnegie Mellon, Ng moved to the Massachusetts Institute of Technology, or MIT, in Cambridge, Massachusetts. There he received his Master of Science degree in 1998. Among the achievements he left behind during his time at MIT, what stands out is not a paper but a piece of software. He built, on his own, a search engine that automatically found and indexed machine learning research papers scattered across the web. Paper search services are nothing unusual today, but in the early days of the internet, a system that automatically gathered and organized scattered academic material was rare in itself. The search engine Ng built was one of the first systems that was publicly accessible and automatically indexed, and it served as a forerunner to CiteSeer (later renamed ResearchIndex), the large-scale academic search engine that would go on to be widely used in academia. No record survives of what prompted him to decide to build this system or what technical obstacles he faced during development. Still, this episode does suggest one tendency in Ng as a researcher. He was someone drawn to turning theory into a tool that could actually be used, rather than leaving it as theory alone. This tendency would recur repeatedly over the decades that followed.

After completing his master's degree at MIT, Ng moved further west to pursue a PhD in computer science at the University of California, Berkeley. His advisor was Michael I. Jordan, a scholar with a worldwide reputation in machine learning. Jordan is regarded as a scholar who built a bridge connecting statistics and machine learning, and several of the students trained under him went on to grow into leading researchers in the field. The committee for Ng's doctoral dissertation included, besides Jordan, Peter Bartlett and Stuart Russell, widely known as the author of a leading artificial intelligence textbook. Given that all three were among the

foremost scholars in their respective areas, one can imagine the level of academic scrutiny Ng's doctoral work had to withstand. No record survives of the specific circumstances that led Ng to choose Jordan as his advisor or of any anecdotes about the scholarly exchanges between the two.

The title of his doctoral dissertation was "Shaping and Policy Search in Reinforcement Learning." Within the mathematical framework of Markov decision processes and partially observable Markov decision processes—used to study how an agent learns optimal behavior in an uncertain environment—the dissertation dug deeply into two problems. The first was the problem of reward shaping. In reinforcement learning, an agent learns by receiving rewards, but if a reward is given only upon reaching the goal, learning can become excessively slow. There had been attempts to speed up learning by adding small rewards at intermediate steps, but such approaches risked leading the agent to find an unintended shortcut and drift away from the original goal. Ng analyzed this problem mathematically and proved that only a specific form of reward function, called potential-based, could accelerate learning without disturbing the originally intended optimal policy. The second was the problem of policy search. In situations where the environment is so complex that it is impossible to know exactly how the state will change, it can be more practical to run a simulator and directly test various policies rather than precisely calculating the value of every single case. To this end, Ng proposed a method called the trajectory tree. To verify this theory, he ran experiments using 5x5 and 8x8 grid-world environments as well as a simulator that mimicked bicycle riding. The content of the dissertation was not confined to abstract equations but was validated through systems that actually worked.

Accounts differ on exactly when he obtained his doctoral degree. One source records that he received the degree in 2002, but the cover of the original dissertation lists the submission date as fall 2003. The surviving material is not sufficient to determine which date represents the final administratively confirmed one. This chapter sets the two records side by side without asserting that either one is correct.

As he was finishing his doctoral studies, in 2002 Ng joined the faculty at Stanford University. He held a joint appointment as a courtesy professor in the Computer Science Department and an assistant professor in the Electrical Engineering Department. The fact that he held positions spanning two departments shows that the subjects he researched did not stay confined to either pure theory or engineering application alone. Once settled at Stanford, Ng went on to become director of the organization commonly known as SAIL, the Stanford Artificial Intelligence Laboratory. SAIL grew into a large academic institution encompassing more than twenty faculty members and multiple research groups, and in leading that organization, Ng had a close view of the process by which the entire field of artificial intelligence broadened and deepened academically.

Among the achievements Ng produced at Stanford, one that stands out is his research on autonomous flying helicopters. A helicopter is by nature an unstable machine. Even a slight imbalance can send the motion of the rotor blades in unpredictable directions, which is why even human pilots require lengthy training to fly one. Getting a computer to control such a machine on its own was, at the time, one of the foremost unsolved problems in robotics and control theory. Ng took on this problem together with students including Pieter Abbeel and Adam Coates. The approach they chose was a method called apprenticeship learning and inverse reinforcement learning. Rather than a person programming optimal rules one by one, this method involved observing flight demonstrations performed by a skilled expert pilot and working backward from those demonstrations to infer what constituted good flying. The research team used a Yamaha R-50 radio-controlled helicopter. They had an expert pilot demonstrate the desired stunt maneuvers roughly ten times, then selected the five smoothest flight trajectories from among them and fed that data into the system. Based on this data, the helicopter learned its own reward function and flight dynamics model, and as a result it came to perform maneuvers on par with an expert pilot on its own, including inverted flight, extreme aerobatics, and autorotation—an autonomous landing performed by falling without engine power. This research stood as an example showing that the theoretical framework of reinforcement learning could work on an actual physical machine as well.

During this period, Ng published more than two hundred papers and established his standing in academia. The subject of reinforcement learning he had taken up at Bell Labs carried through his doctoral dissertation at Berkeley and on into the helicopter project at Stanford, and this academically grounded foundation became the starting point for the many forks in the road he would later walk. He would go on to serve as the founding lead of the Google Brain project between 2011 and 2012, to co-found the online education platform Coursera building on the online machine learning course he had taught at Stanford, and further to work as chief scientist at Baidu in China. Each of these choices will be taken up again in the chapters that follow. What this chapter must make clear, however, is that behind every one of those choices lay the twenty-six years spanning his birth in London, his upbringing in Hong Kong and Singapore, and his passage through Carnegie Mellon, MIT, and Berkeley on the way to Stanford. Almost no personal feelings or private stories survive from this period. What remains are only traces bearing the names of degrees, papers, laboratories, and experimental results. Yet even from those traces alone, the path by which one person grew into a world-class artificial intelligence researcher emerges clearly enough.

That a child was born into a family of immigrants who had moved from Hong Kong to London was, in those days, one of the common scenes of British society. In the latter half of the twentieth century, Britain took in a considerable number of migrants from its former colony of Hong Kong, and many of them settled in Britain for the purpose of study or employment. Andrew Ng's parents likely settled in London as part of this same flow, but no

surviving record confirms the specific circumstances that led them to emigrate or what kind of life they built in London. Still, the very fact that he was granted the status of a British National (Overseas) forms one continuous thread with the later process by which he became a U.S. citizen and came to be called Hong Kong American. The fact that the country of one's birth, the country one grows up in, and the country whose citizenship one obtains as an adult can each be different does not seem unrelated to the career he would later build, moving between the United States and China for work. There is no basis to conclude outright that a cross-border upbringing translated directly into an international sensibility, but it is not unreasonable to guess that, at the very least, he felt no particular psychological barrier as an adult moving between schools and companies in different countries.

Raffles Institution in Singapore is an institution whose standing cannot be adequately captured by simply calling it a prestigious high school. Founded in the early nineteenth century by Stamford Raffles, the school has formed the backbone of Singapore's education system ever since, and even after independence it remained central to the Singaporean government's policy of identifying outstanding talent early and educating them intensively. That Andrew Ng graduated from such a school shows that he was already classified as a top-tier student within the Singaporean education system in his teens. Singapore is a country that has consistently ranked near the top in international assessments of mathematics and science education, and the path from its top schools to elite American engineering universities was not uncommon even then. What subjects Ng himself particularly excelled in at this school, or what led him to choose Carnegie Mellon specifically, remains unconfirmed. This period of career decisions, then, survives only as administrative traces such as transcripts and admission records, never explained in his own words.

Looking more closely at the combination of triple majors he chose at Carnegie Mellon reveals that this choice was not simply a matter of accumulating easy credits. Double majoring is common at American universities, but completing three majors at once, spanning entirely different academic domains, requires satisfying all the required coursework demanded by each department. The computer science department required courses such as algorithms, data structures, and programming language theory; the statistics department required probability theory and inference theory; and the economics department required microeconomics and econometric methodology. These three groups of courses overlap in places while also demanding different kinds of thinking, so for a student, simply planning each semester's coursework required considerable effort in itself. The fact that Ng bore this load and still graduated first in his class is a confirmed result showing that he was already adept, from his undergraduate years, at moving fluidly across multiple fields. This ability would recur repeatedly throughout his later research career. The field of machine learning itself sits at the intersection of mathematics, statistics, and computer engineering, and the problems Ng would go on to tackle likewise often moved between pure theory and practical application.

To describe the standing of his doctoral advisor Michael I. Jordan a little further, he is regarded as a scholar who built the theoretical framework connecting statistics and machine learning at a time when the two fields did not yet speak to each other. Several students who passed through Jordan's lab went on to lead machine learning research at their own universities or companies, and against this backdrop his lab was known as an especially competitive place even within Berkeley at the time. Peter Bartlett, who sat on Ng's dissertation committee, was a scholar known for rigorous mathematical analysis in statistical learning theory, and Stuart Russell was the author of a textbook that systematically organized the entire field of artificial intelligence—so influential that nearly every student entering the field first learned artificial intelligence through his book. The fact that these three people examined a single doctoral dissertation gives some sense of the level of scrutiny Ng's research had to pass. A dissertation defense is typically conducted before a committee made up of one advisor and several professors from within and outside the department, with each member probing the dissertation's weaknesses from a different angle—so the composition of the committee itself reflects the academic weight the dissertation had to bear.

The two titles Ng held at Stanford after completing his doctorate—courtesy professor in the Computer Science Department and assistant professor in the Electrical Engineering Department—meant that he occupied a position where he could research across two departments without belonging entirely to either one. Such dual appointments are not uncommon at American research universities, but because funding structures and student advising practices differ depending on which department one belongs to, a new professor obtaining a position spanning two departments also means that both departments judged his research to be needed. From the computer science department's perspective, Ng was a researcher in artificial intelligence and machine learning, a field whose importance was already growing at the time; from the electrical engineering department's perspective, he was a researcher versed in signal processing and control theory who could contribute to problems related to robotics and automatic control. At the point where the demands of these two departments overlapped, Ng expanded the scope of his research.

Looking more closely at the scale of SAIL during the period he led it, the fact that more than twenty faculty members belonged to it shows that this laboratory was an academic community that had grown beyond the level of a single lab. That he came to lead a research organization of such considerable scale within Stanford at a young age is also evidence that he had already earned substantial trust within academia. The position of laboratory director is not simply about continuing one's own research; it also requires coordinating multiple research groups moving in different directions, placing incoming graduate students and postdoctoral researchers, and explaining the direction of the laboratory as a whole to outside parties. No record survives of the administrative burdens or organizational details Ng would have faced in taking on this role, but given his later history of founding the organization Google Brain,

co-founding the company Coursera, and leading a large organization at Baidu, one can guess that his experience at SAIL was where he cultivated a sense for organizing people and resources.

The figure of more than two hundred papers is itself an indicator of a researcher's productivity. Producing a single academic paper typically requires defining a problem, reviewing relevant prior research, designing experiments, analyzing results, and passing peer review. The fact that he repeated this process more than two hundred times shows that Ng continuously pursued multiple lines of research at once from the time he settled at Stanford onward. The subjects he worked on were not limited to reinforcement learning and robotics but extended into computer vision, natural language processing, and large-scale neural network training, and this broad research history shows that he was not tied to any single technology but kept adjusting his research direction to follow the changes across the entire field of artificial intelligence. As his helicopter research showed, he was a researcher who took particular care to confirm whether theory actually worked on a real machine, and this attitude shows up not only in the number of his papers but also in the number of actually functioning systems he produced.

Looking at Ng's career during this period as a whole, the time it took for a child born in London to pass through Hong Kong and Singapore, cross three American universities, and settle at Stanford came to less than thirty years. Within this short span, he earned his bachelor's, master's, and doctoral degrees at three different universities, worked as a researcher at a world-renowned telecommunications laboratory, and became a professor at an elite university and director of a large research laboratory. No document confirms the personal motivation or psychological drive behind this trajectory, but the surviving record of degrees, papers, and laboratory leadership alone makes clear how quickly and how broadly he expanded his place in the academic world. What he had built up to this point was not yet a name widely known to the public, but a position of trust among fellow researchers. That trust would soon call him out onto a wider stage beyond the laboratory.

There is good reason why Carnegie Mellon University came to be called a cradle among artificial intelligence researchers. In 1965, the school became the first American university to establish an independent computer science department. At its center were two scholars, Herbert Simon and Allen Newell. Working from the premise that a computer could imitate the process of human thought, the two built the framework of early artificial intelligence research and jointly received the Turing Award, computer science's highest honor, in 1975. Built on the foundation laid by these scholars, Carnegie Mellon established itself over the following decades as one of the centers of robotics and artificial intelligence research, and by the early 1990s, when Ng enrolled, it had already become a school drawing outstanding students from around the world. The school's circumstances also underlay his ability to choose a triple major.

At a time when universities offering a broad curriculum spanning computer science, statistics, and economics were uncommon, Carnegie Mellon already had the scale and structure to support an education that crossed multiple disciplines.

The place of the paper search system he built by hand during his MIT years becomes clearer when set against the academic circumstances of the time. In the late 1990s, the internet was spreading rapidly, but academic papers were still managed individually, scattered across the websites of various conferences and journals. To find a relevant paper, a researcher often had no choice but to search through multiple sites one by one or ask a colleague. In this situation, systems that automatically crawled the web to gather and index papers were rare, and the fact that a student fresh out of undergraduate studies designed and built such a system on his own shows that the technology he was working with did not remain confined to theory. CiteSeer, the system that later succeeded this project, went on to be used by numerous universities and research institutions to track paper citation relationships and remained one of the basic tools of academia for a long time. Services such as Google Scholar and Semantic Scholar have since taken over this role, but before such services existed, academia relied heavily on early systems like the one Ng built.

The period when Ng pursued his doctorate at Berkeley was also a time when the field of artificial intelligence itself was changing direction. In the 1970s and 1980s, artificial intelligence research went through a period in which results repeatedly fell short of expectations, and funding and interest declined sharply. This period is commonly called the AI winter in academia. It became clear that an approach in which people programmed in rules and logic one by one could not handle the complex problems of the real world. Passing through this winter, researchers turned away from hand-coding rules and toward finding statistical patterns in data instead. It was around this time that probability, statistics, and optimization theory began to establish themselves as the language of artificial intelligence research. Ng's advisor Michael I. Jordan was a scholar who, precisely during this turning point, worked to connect the language of statistics with the problems of artificial intelligence. The reinforcement learning that Ng worked on during his doctorate was likewise a methodology grounded not in rule-based approaches but in a statistical, probabilistic framework of trial, error, and reward—so his research subject itself sat right on the line of this transition.

The history of SAIL, the Stanford Artificial Intelligence Laboratory, also helps gauge the weight of the position Ng took on. The laboratory was founded in 1962 by John McCarthy, one of the scholars who coined the term artificial intelligence, making it one of the oldest artificial intelligence research institutions in the world, established directly by its own founder. In its early years, SAIL made its name through experiments such as object manipulation using robotic arms and autonomous mobile robots, and it continued to produce steady research in computer vision, natural language processing, and robotics in the years that followed. Coming

to lead a laboratory with this history meant that Ng had come to occupy a position representing an organization within the artificial intelligence research community, not merely one of being recognized as an individual researcher.

The year 2002, when Ng was appointed a Stanford professor, was also a turbulent time for Silicon Valley as a whole. The dot-com boom, which had peaked in early 2000, began collapsing rapidly the following year, and countless internet companies shut down or scaled back sharply. The venture capital and startup ecosystem surrounding Stanford also contracted during this period. Amid this situation, the university offered a relatively stable base for research, and Ng was able to build his academic achievements at SAIL somewhat apart from the ups and downs of industry. But this contraction in Silicon Valley did not last long. From the mid-2000s on, as search and online service companies grew again, Silicon Valley regained new vitality, and the channel through which technology developed in university labs moved into industry gradually widened. This flow within Silicon Valley formed the backdrop against which Ng would later form ties with Google and put his research to the test in industry beyond the university.

An Early Researcher: Robots and Reinforcement Learning

The helicopter came first. Before the cat in the YouTube videos, before the online classroom that would draw students from around the world, Andrew Ng was teaching a robotic helicopter how to flip itself over in midair. After he had settled in as a professor at Stanford, the field that first brought his name to public attention was not large neural networks but reinforcement learning — a way of training a robot or software agent to discover better actions on its own through trial and error, colliding with its environment without ever being handed the right answer in advance. The research from this period drew none of the fanfare or public attention that would come later, but it was the period that gave him his instinct for how to work with data at scale.

Andrew Ng earned his doctorate at UC Berkeley under Michael I. Jordan, a world-renowned scholar in machine learning. His dissertation committee also included Peter Bartlett and Stuart Russell. His doctoral thesis dealt with the mathematical framework known as the Markov Decision Process, or MDP, and its extension, the Partially Observable Markov Decision Process, or POMDP. Both concepts still serve as the backbone of reinforcement learning theory today. When an agent can fully know what state it is currently in, the problem is called an MDP; when it can only guess at that state, it is called a POMDP. Real-world situations in which a robot must make decisions while viewing the world through a single camera are, for the most part, closer to the latter.

One of the dissertation's contributions concerned a topic called reward shaping. Researchers eager to speed up learning often feel the temptation to hand-tune small extra rewards whenever an agent gets closer to a goal. The trouble is that such tampered rewards can lead the agent to discover shortcuts that diverge from the original objective. If a robot is given points every time it moves closer to a goal, for instance, it may learn to loiter near the goal collecting points indefinitely instead of actually advancing toward it. Andrew Ng dug into this problem mathematically and proved that if rewards are reshaped only in a specific form — a potential-based reward shaping function — the original optimal policy remains unchanged. What might look like a narrow, technical theorem on the surface became a standard for judging when a researcher's hand-tuned reward design is safe and when it is dangerous.

Another contribution was the trajectory tree and the PEGASUS algorithm. In complex POMDP environments, computing the exact value of every state and action is often simply infeasible in practice. Rather than solving for a perfect value function, he devised a method that ran a simulator many times and stitched the resulting trajectories into a tree structure to estimate a policy's performance. The PEGASUS algorithm went a step further, converting a simulation problem that fluctuates stochastically into a deterministic form by fixing the

random seed. When results differ from one simulation run to the next, it becomes difficult to compare which policy is truly better. Fixing the random numbers allows policies to be compared fairly under identical conditions. This methodology was first validated in simple test environments such as 5x5, 8x8, 10x10, and 50x50 grid worlds, but behind it lay the groundwork for tackling far more complex physical systems.

After moving to Stanford, Andrew Ng's research team set out to carry this theory over into real machines. Working with colleagues including Pieter Abbeel and Adam Coates, they chose an autonomous helicopter as their target. Helicopter flight is considered one of the classic hard problems in control theory. The dynamics of the aircraft are high-dimensional, asymmetric, noisy, and nonlinear, and they even exhibit non-minimum-phase dynamics, in which the direction the aircraft momentarily moves diverges from the direction the pilot pushes the stick. There had been earlier attempts to fly a helicopter through aerobatic maneuvers by hand-coding target trajectories one by one, but most of them hit their limits when faced with intricate maneuvers.

The path the research team chose was apprenticeship learning: rather than having a person hand-craft trajectories, they had the robot watch a skilled pilot's demonstration flights and learn from them. First they gathered twenty minutes of flight data and used linear regression to build a basic dynamics model, one that predicts roughly how the aircraft will move given a particular stick input. Next, a human expert flew the same aerobatic maneuver roughly ten times as a demonstration. Even a person cannot trace exactly the same trajectory every single time. The research team used a statistical technique called the EM algorithm to filter out the pilot's mistakes and wobbles scattered across the multiple demonstration flights and infer, in reverse, the ideal trajectory hidden behind them. Rather than simply copying the human's imperfections, they mathematically reconstructed the flawless maneuver the person had originally intended to perform.

To translate the target trajectory obtained this way into an actual flight, the research team used a receding-horizon differential dynamic programming (DDP) controller operating at a 20 Hz cycle, along with a Gauss-Newton LQR controller. The experiments used remote-controlled helicopter airframes called the XCell Tempest and the Synergy N9. To precisely measure the aircraft's attitude and orientation, they mounted a MicroStrain 3DM-GX1 sensor, and multiple ground-based vision cameras tracked the aircraft's position through triangulation. Data newly gathered during autonomous flight was fed back into the model, used to locally refine the dynamics model for a specific trajectory. Repeating this process many times over, the model and the controller grew steadily closer to the actual behavior of the real aircraft.

The result was the world's first autonomous success at a series of highly demanding aerobatic maneuvers: the flip, in which the aircraft rotates in place; the roll, in which it turns sideways; the loop, in which it traces a circle; the tic-toc, in which the aircraft swings back and

forth like a pendulum with its tail pointed down; the hurricane, in which it traces a cone-shaped path while flying backward at high speed; and the chaos, a continuous series of vertical rotations. Among these, the tic-toc and the chaos were maneuvers that hand-crafted trajectory methods had rarely managed to pull off. The precision of the autonomous flight rose to a level that surpassed even skilled human pilots. In an autonomous airshow flight, the positional error, which started at 3.29 meters, dropped to 1.75 meters after the data gathered from autonomous flights was repeatedly folded back into the model. A cycle had taken hold in which the experience the aircraft gained by flying on its own made the aircraft itself more accurate in turn.

The most dangerous part of this research was the autorotation landing, attempted with the engine completely off. When a helicopter's engine cuts out, the rotor no longer receives power, but the airflow generated during descent can still keep the rotor spinning, and that rotational energy can be used for landing in the final moment. It is a hazardous maneuver that requires training even for a real human pilot. The research team designed the descent to hold the rotor speed at 1150 RPM, lower than the 1700 RPM used in ordinary powered flight, while maintaining a forward speed of 8 meters per second and a descent speed of 5 meters per second. At 9 meters above the ground, the aircraft began a flare maneuver, raising its nose to slow down, and executed precisely controlled deceleration to bring its speed to nearly zero by the time it reached 0.5 meters above the ground. Using this method, the team attempted a total of twenty-five autorotation landings and succeeded in every one without a single instance of airframe damage.

If the helicopter aerobatics were an experiment that tested the theories of reinforcement learning and apprenticeship learning under extreme conditions, the research team also reached, during this same period, toward indoor robots meant to work alongside people. They built a low-cost robotic arm to replace expensive industrial robot arms, using a series-elastic design and stepper motors, achieving a payload of 2 kilograms, a top speed of 1.5 meters per second, and a repeatability of under 3 millimeters. With this arm they even succeeded in a cooking demonstration, pouring and flipping pancake batter safely enough not to injure a person nearby. In the Stanford AI Robot project, known as STAIR, a robot equipped with a 7-degree-of-freedom arm and a Barrett robotic hand was used to build an autonomous checkout clerk. Using nothing more than a single depth image captured by a 3D sensor, this robot could grasp an object it had never seen before, rotate it in its hand, locate the barcode, and read off the numerical code.

Research also continued into sharpening the tactile sense at the robot's fingertips as it grasped objects. The team built its own miniature optical proximity sensors to attach to the fingertips of the Barrett hand, allowing the hand to detect which way a surface was oriented before making contact and to build a controller that adjusted its grasp point in real time.

Beyond this, work was also done on recognizing and operating elevator button panels, driven by the realization that a robot navigating an indoor space fully on its own would need to be able to take an elevator by itself. With the four-legged robot LittleDog, research was carried out to use stereo vision alone to perceive the terrain ahead in three dimensions, allowing the robot to cross uneven, irregular obstacle-strewn ground. The targets differed from one project to the next — a helicopter, a robotic arm, a checkout robot, a walking robot — but running beneath all of them was a single shared direction: making the robot learn from data on its own, rather than having a person hand-code every rule.

This research aimed at making a robot's eyes smarter naturally ran into a wall. To improve object recognition, a person ultimately had to hand-design which features the system should look at to make its judgments, and this approach carried the limitation that the design work had to be redone every time the robot encountered a new object or a new environment. The research team turned instead to large-scale unsupervised learning, in which, rather than a person predefining features, a vast quantity of unlabeled data would be shown to the system so that it could discover the features on its own. The first obstacle this shift ran into was computational speed. Methods promising at the time, such as deep belief networks and sparse coding, were theoretically appealing but painfully slow — training tens of millions of parameters on a single central processing unit could take weeks. The research team pioneered the use of graphics processing units, or GPUs, to break through this bottleneck — the idea of using hardware originally built for rendering images on a screen to train large neural networks instead. A GPU could process an enormous number of computations in parallel at once, cutting training time that would otherwise take weeks down to a matter of days.

Once the computational bottleneck was resolved, the next goal shifted to how large a neural network could be scaled up. The research team proposed a methodology for extracting high-level features from vast quantities of unlabeled data. They built an objective function based on TICA — Topographic Independent Component Analysis — including a reconstruction cost, and used it to train the weights corresponding to two layers of the neural network. Without ever being told the correct answer for what object to look for, the system, simply by scanning through a vast amount of data, developed the ability to recognize shapes such as human faces or cats' heads on its own. In supervised learning, a person must label each image one by one — this photo contains a cat, this one does not — but this experiment showed that similar results could be reached through sheer volume of data, without any labeled answers at all.

It was a progression that began with reinforcement learning, passed through the extreme control of a helicopter, moved through a series of experiments handling a robot's hands and feet, and finally arrived at breaking through the computational bottleneck with GPUs and letting unsupervised learning discover features on its own. Each piece of research was

published as its own independent paper, but behind all of them lay a single direction: a shift away from a world where people define the rules, toward one where data generates the rules itself. This experiment, which trained tens of millions of parameters on GPUs to discover faces and cats' heads on its own, became, a few years later, the bridge to the Google Brain project, in which a neural network with more than a billion parameters learned to recognize cats in YouTube videos. What Andrew Ng built up as an early researcher working with robots and reinforcement learning was not a dazzling result but a period in which he learned, through direct experience, how data and computation had to be handled for a system to grow smarter on its own.

There was a reason the helicopter aerobatics drew unusual attention within academia. Control theory researchers had long used helicopters as a proving ground for system control. Unlike an airplane, a helicopter can hover in place even at low speed and can freely change direction, but that same ability makes its motion difficult to predict, since the rotational forces and reaction torque generated by the rotor become entangled with the effects of wind. Because the direction the aircraft actually responds momentarily diverges from the direction the stick is pushed, the stable system models used in traditional control theory struggled to account for extreme maneuvers like aerobatics. Earlier attempts to fly a helicopter autonomously by hand-crafting trajectories had managed hovering and gentle turns, but ran into a wall whenever maneuvers combined rotation with reversal. Against this backdrop, the results Andrew Ng's research team produced took on a problem that had long resisted solution in both robotics and control theory.

The concept of apprenticeship learning at the core of this research also deserves closer examination. In ordinary reinforcement learning, the researcher must define the reward function in advance. Learning can only begin once a person has specified in numbers what score to award for a given action and what penalty to impose for another. But expressing, in a single number, the grace of a helicopter's rotation or the feel of a smooth landing turns out to be surprisingly difficult. A human pilot can tell good flying from bad by feel, but translating that judgment into an explicit formula is not easy. Apprenticeship learning solved this problem in reverse. Rather than having a person directly set the reward function, it observes how a skilled pilot actually flies and infers, in reverse, the goal that lies behind it. Even if a person's demonstration flight is imperfect, statistically filtering out the ideal trajectory common across multiple attempts made it possible to transfer even a sense that a person cannot put into words into something a machine could learn. This approach later became a methodology repeatedly used across robotics, autonomous driving, and other fields to solve problems where designing a reward by hand is difficult.

The lab at Stanford that led this research bound together several directions of work under a single name: STAIR, the Stanford AI Robot project. The team working on helicopter flight, the

team grasping objects with a robotic arm, and the team building the checkout-clerk robot were, on the surface, solving different problems, but in fact they collaborated within the same lab, sharing people and equipment. Pieter Abbeel took on the role of refining the theory of apprenticeship learning in the helicopter aerobatics research, and later expanded his research into grasping and manipulation problems at the Stanford robotics lab. Adam Coates, after participating in the helicopter control experiments, moved on to large-scale unsupervised learning research and went on to work on training neural networks with GPUs. While one person worked on the robotic arm, another flew the helicopter, and yet another refined barcode recognition for the checkout robot, they shared results with one another and together cultivated a shared attitude: building systems that learn from data. One could say that the mathematical rigor cultivated in Michael Jordan's Berkeley lab and the hardware experimentation culture that grew in the Stanford lab converged within a single person.

There was also good reason behind the sequence of first validating the theory in a simplified grid environment called a grid world, and only then applying it to an actual helicopter — an expensive, dangerous machine. A grid world is a virtual checkerboard-like space in which a robot moves only up, down, left, or right. In such a small environment, one can quickly and safely confirm whether an algorithm behaves as the theory predicts and whether policy search actually finds better actions. Had the team started experiments directly on a real helicopter and the algorithm turned out to have a flaw, the result could have been a crashed aircraft and destroyed, expensive equipment. Only after repeatedly validating the PEGASUS algorithm across grids of various sizes, from 5x5 to 50x50, could the research team extend the same principle — using twenty minutes of flight data, the EM algorithm, and a receding-horizon DDP controller — and apply it to a real aircraft. This procedure of cautiously carrying a principle validated on a small problem over to a larger one was an attitude that would recur again and again in the research Andrew Ng would go on to pursue.

A similar concern ran through the research on robotic arms grasping objects. For a checkout-clerk robot to grasp an object it had never seen before, a computer had to convert an image captured by a camera or depth sensor into numbers it could understand — where the object was and what shape it had. These converted numbers are called features. For a long time, the standard approach was for a human researcher to define, rule by rule, information such as where an object's edges lay, which way its surface was tilted, and how its colors were distributed, and hand all of that to the computer. The problem was that these rules only worked well for particular objects or environments. Whenever a new product appeared on the checkout counter or the lighting changed, the pre-crafted rules frequently broke down. The research using a fingertip-mounted proximity sensor to detect surface orientation in advance and adjust the grasp point in real time was, in the end, also an attempt to compensate with sensors for variables that rules set by people alone could not handle. The research on recognizing elevator button panels and LittleDog's terrain recognition research belonged to

the same context. If a person had to redesign the rules every time the robot encountered a new space or a new object, it was hard to call the robot truly autonomous. The more the team tried to make the robot's eyes and hands smarter, the more the limits of human-designed rules kept resurfacing.

This experience became the catalyst that pushed the research team toward having features learned from data on its own, rather than defined by a person. The TICA-based objective function trains its weights using reconstruction cost — that is, how closely the neural network can restore an input image to its original form — as its criterion. Even without a person telling it that this photo contains a face or that photo contains a cat, the neural network, by scanning through a vast quantity of photographs, captures on its own the shapes that recur again and again. This principle itself was not a new idea; methods such as deep belief networks and sparse coding were already known in academia. But running such methods at a scale useful in practice required repeatedly adjusting tens of millions to hundreds of millions of weights, and with a single central processing unit of the kind commonly used at the time, this computation took weeks. That is why the research team's choice to bring in graphics processing units mattered. This hardware, originally designed to render three-dimensional graphics on screen, was optimized for processing an enormous number of small computations simultaneously. Since training a neural network was, in the end, also a repetition of multiplying and adding vast numbers of figures at once, hardware built for rendering graphics turned out, unexpectedly, to be well suited to training neural networks too. As computation time shrank from weeks to days, and then from days to a single day, the research team was able to experiment with neural networks at a scale they had never dared attempt before.

In the computer vision academia of the time, using human-designed features was still the standard approach. Research that discovered shapes such as faces or cats' heads from data alone, with no labeled answers, was not immediately embraced widely by the conference community. But what this experiment demonstrated was clear: without a person telling it exactly what to look for, a sufficiently large neural network shown a sufficiently large amount of data could discover meaningful patterns on its own. This result was met with a skeptical eye even within academia. Neural networks as a methodology had, for a time, been treated as a fashion that had passed, and the approach of solving problems by scaling up data and computation looked, to researchers who preferred building elegant mathematical models, like a somewhat crude solution. Even so, Andrew Ng's research team carried the attitude they had confirmed in the helicopter experiments — letting data find the answer on its own rather than having a person spell it out — directly over into computer vision.

The weight carried by the autorotation landing experiment is also worth revisiting. Landing safely on the ground using only the rotational energy of the rotor, with the engine off, is an emergency procedure that requires long training and nerve even for a real pilot. The fact that

the research team attempted this landing twenty-five times and completed all of them without a single instance of airframe damage showed that the learned policy did not merely work under calm conditions but performed consistently even in an extreme situation with no power at all. This experiment, meticulously designed down to the rotor speed, the descent speed, and the altitude at which the flare maneuver began, was also a venue for verifying whether a policy learned from data could be trusted to meet the safety standards that people set. The sense that this kind of repeated verification must follow before a learned system is sent out into dangerous situations beyond the laboratory became a shared standard in later research dealing with robots and autonomous systems.

Looking back, the research from this period dealt with different machines and different problems, but shared a common experimental procedure: formulate the problem mathematically, verify that the formulation holds in a small, safe environment, and only then move it over to real hardware to confirm the results. The helicopter aerobatics work began with a grid world and advanced to a real aircraft; the robotic-arm research filled in the limits of human-set rules with sensors and data; the unsupervised learning research broke through the computational bottleneck with graphics processing units. These three directions of work were published as separate papers, but they were a single experimental culture built up by the same people moving among them within the same Stanford lab. This attitude — of letting data discover the rules on its own rather than defining the answer in advance — became the natural passageway into the next period of work, one that would go on to handle far larger neural networks and far vaster quantities of data.

Unlike the helicopter's motion, which curves and winds like a mountain ridge, reinforcement learning theory broadly has two lines of approach. One assigns a value to each state and action and picks whichever has the largest value — value function estimation. The other changes the policy itself without separately computing values, searching for the direction in which performance improves — policy search. The path the PEGASUS algorithm chose was the latter. In a complex POMDP, precisely assigning a value to every single state was often more than the computation could bear. So rather than computing values, the research team chose instead to run a policy through the simulator many times and compare the resulting trajectories to select the better policy. The concept of potential-based reward shaping is bound up with this same choice. The potential referred to here is similar to the idea, in physics, of the potential energy held by an object at a height. If a number is assigned in advance representing how close a given state is to the goal, and a reward is added or subtracted only by the difference in that number each time the robot moves to a new state, then no matter how much the robot winds and circles around, the ranking of what was originally the optimal path remains undisturbed. It is no different from the arithmetic in which, as long as the height difference between the starting point and the destination is the same, the potential energy gained in the end is the same no matter which path is taken up the slope.

There is one more point worth noting in the research on robotic arms grasping objects. Among the work carried out alongside the checkout-clerk robot research was a study on learning multiple simultaneous contact points where several fingers touch at once. When a human hand grips a cup or a bottle stably, a single fingertip touching it is not enough. Two or more points must press the object from different directions for it to be held firmly without slipping or rotating. In carrying this principle over to the robotic hand, the research team tried a method that learned combinations of multiple contact points from image data of the object itself, rather than having a person define, rule by rule, which finger should go where to grip stably. Since every object has a different shape, and the number of possible combinations of graspable points is essentially limitless, it was simply not possible for a person to hand-craft rules covering every case in advance. The trained model could, on its own, select a combination of contact points that would grip stably based on the surface shape, even for an object it had never seen before. Since the checkout-clerk robot needed to be able to rotate an object every which way without dropping it in order to find its barcode, this learning of stable contact-point grasping was close to the very foundation on which the entire checkout robot depended.

There was also intent behind the name given to the STAIR project. The helicopter team, the robotic-arm team, the checkout-clerk team, the elevator-recognition team, and the walking-robot team each produced results in separate papers, but they were bound together under the single umbrella of STAIR, because the team had set a goal that a single robot ought to be able to do all of it — move about a room on its own, pick up objects, ride an elevator to another floor, and find and read the barcode of an object it had never seen before. Whether it was the legs of LittleDog walking indoor terrain, the hand of the checkout-clerk robot, or the arm pressing an elevator button, each of these capabilities was, in the end, one piece needed for a single robot to move autonomously through spaces where people live. The reason different teams, experimenting with different hardware, nonetheless shared a methodology of learning from data was that the problem they were ultimately trying to solve was, from the outset, a single connected goal.

The attitude running through the research of this period was the principle of verifying with something small before tackling something large. That is also why the PEGASUS algorithm was tested repeatedly across grids of various sizes, from 5x5 up to 50x50. Each time one side of the grid grows, the number of possible states the robot could be in expands by the square of that growth. Only after first confirming that a method which worked in a small grid did not collapse in a much larger one could the research team carry that principle over to a helicopter — an expensive, dangerous machine. This same sequence recurred identically when the team moved on to learning contact points for the robotic arm and to large-scale unsupervised learning. Establishing a principle on a small problem, confirming that the principle holds as the problem grows larger, and only then moving on to the next scale — this sequence became,

years later, the very attitude carried into the work of scaling up both the size of neural networks and the volume of data together. This period, spent handling a robot's hands, feet, and eyes one at a time, was, in the end, also a training ground for confirming, through direct experience, the limits of cautiously expanding the scale of a problem.

Google Brain and the Scaling Experiment

Inside Google in 2011, there was one strange team. It was not a division that carried the company's revenue, like search or advertising, nor was it the kind of research lab commonly imagined, one that only produces papers. It was a team willing to bet the company's resources on a technology that had, at the time, stood a step outside the mainstream of academia: neural networks. That team was called Google Brain, and the person who founded and led it was Andrew Ng.

The beginning traces back to 2010. Andrew Ng, then a Stanford professor, sent a bold proposal to Google's leadership. It called for training neural networks using enormous computing resources and vast amounts of data. Google accepted the proposal. The following year, in 2011, Andrew Ng founded Google Brain, a deep learning project, and took charge of it.

At the time, focusing on making models bigger was an unfamiliar approach within the AI academic community. Neural network research had long been regarded as a field where the sophistication of algorithms mattered more than scale, and the idea of solving problems by piling up mountains of computers was often dismissed as an inelegant attempt. Some fellow researchers even warned Andrew Ng that pouring effort into massive neural networks and scaling would not help his career as a researcher. But the first and only goal Andrew Ng set for the Google Brain team was clear: grow the neural network, and pour data into it.

Where did this conviction come from? At its foundation lay a hypothesis drawn from how the human brain works, the so-called single learning algorithm hypothesis. Before starting Google Brain, Andrew Ng had read mountains of neuroscience papers and spent time among neuroscientists, trying to figure out how the human brain operates in order to find clues for better AI algorithms. But after confirming that neuroscience at the time had barely uncovered the brain's operating principles, he gave up on the path of directly applying neuroscience to building intelligence. Even so, the generality of the biological brain remained deeply lodged in his thinking.

This generality means the following. Human DNA is short, and the amount of information it can encode is limited. And yet humans can carry out an enormous range of entirely different tasks, from earning a doctorate in mathematics to riding a motorcycle to answering calls at a call center. The reason is not that the brain is born with dedicated parts prepared in advance for each task. It is because the brain is born with a single engine of adaptability, a single learning engine, capable of specializing itself through experience and training. Andrew Ng believed artificial neural networks could work the same way. Rather than an elaborate design tailored to a specific problem, he believed that given sufficient data and sufficient computing resources, a single neural network architecture could achieve a generality comparable to the human brain.

This belief soon led to a single question. Just as a baby learns the concept of a face on its own without anyone teaching it, could a machine likewise learn the concepts of the world on its own from data alone, with no answer key provided? The Google Brain team decided to turn this question into an experiment. It was an experiment to check whether a feature detector that responds selectively to a specific object at a high level could be trained using only data with no answer labels at all. Here, a feature detector refers to a neuron inside a neural network that detects a distinct whole object, such as an entire face or an entire cat, rather than parts within an image like eyes, nose, or ears. In brain science, a neuron that responds only to one specific object like this is called a grandmother cell, a term drawn from the hypothesis that somewhere in the brain there exists a neuron that fires only when seeing one's grandmother's face. The question the Google Brain team posed was whether a phenomenon like this grandmother cell could emerge inside an artificial neural network through unsupervised learning alone, with no supervision or reward.

To answer this question, the researchers built a dataset of 10 million images randomly extracted from YouTube videos. Each image was a 200x200 pixel color image. The model was a locally connected sparse autoencoder made up of nine layers, with L2 pooling and local contrast normalization applied. An autoencoder refers to a neural network architecture that learns the core structure of data by repeatedly compressing input data and then reconstructing it on its own. This neural network had one billion connections, or parameters. Considering that the large neural networks reported in the world at the time generally had around ten million parameters, this was more than ten times that scale. Of course, compared to the number of neurons and synapses in the human visual cortex, it was still a mere one-millionth in size.

Training a neural network of this size required computing power to match. The research team ran a cluster of 1,000 machines with 16,000 cores for three days. To handle computation at this scale, the team built a new software framework called DistBelief. It implemented model parallelism, a method of dividing neurons and weights across 169 machines, each equipped with 16 cores. On top of this, they combined data parallelism, training five replicas of the core model through 256 parameter server partitions using asynchronous stochastic gradient descent. In other words, they split a single neural network across multiple computers, trained several replicas simultaneously on different pieces of data, and had them continuously exchange results with one another. This method, now taken for granted as common sense in large-scale model training, was at the time a novel design that Google Brain forged through direct trial and error.

The results shook the academic world. When classifying images containing human faces against randomly mixed distractor images, a single neuron had spontaneously formed inside the network that responded selectively only to human faces. This neuron's accuracy was 81.7

percent. What was interesting was the next experiment. Using an image processing tool called OpenCV, the researchers artificially removed images containing human faces from the training data and ran the same experiment again. The neuron's performance then dropped sharply to 72.5 percent, the level of a simple linear filter. This result showed that the network had not acquired such a neuron by chance, but had genuinely learned the concept of a face that actually exists within the data.

This neural network did not stop at human faces. Neurons that had spontaneously learned the concepts of cat faces and human bodies also emerged alongside it. In an experiment using only the left image from stereo grayscale images, the neuron detecting human bodies achieved 76.7 percent performance and the neuron detecting cats achieved 74.8 percent. Both figures far exceeded the comparison groups based on simple linear filters or random weights. The learned top-level features showed stable properties, remaining unshaken by shifting the image sideways, rotating it slightly out of frame, or enlarging or shrinking it.

The researchers did not stop there. They placed a logistic classifier on top of the feature representations learned in this way and took on the task of recognizing the 20,000 object categories of ImageNet. After a week of fine-tuning on 2,000 machines, they recorded 15.8 percent accuracy. This was a 70 percent relative improvement over the best performance at the time. This experiment later came to be known to the world as the YouTube Cat. The news that a neural network shown only YouTube videos, with no answer key, had figured out the concept of a cat on its own was an interesting story even to people outside the research community, and it became one of the events that first introduced deep learning to the mass media.

Around this time, Andrew Ng was still teaching students at Stanford, and preparations for the online course and Coursera that would later emerge to the world were also underway in the same period. The Google Brain laboratory, the Stanford classroom, and the groundwork that would later lead to Coursera were all running in parallel over the same few years.

The YouTube Cat experiment proved the academic potential of large-scale deep learning. But that alone could not convince the entire company. Both inside and outside Google, skepticism about the practical usefulness of deep learning still remained. There was a wide gap between an interesting result from the laboratory and embedding that technology into an actual product that generates revenue. To pull this technology out of the lab and turn it into a capability for the whole company, Andrew Ng devised his own strategy of persuasion. The principle he chose was this: rather than aiming for the single most valuable-looking project that could instantly change the company's strategic direction, he chose to pursue projects that would actually succeed in their first few attempts.

Following this principle, the place he chose as Google Brain's first internal customer was Google's speech recognition team. At the time, speech recognition did not occupy as

important a position in the company's bottom line as applying AI to web search or advertising, which drove revenue significantly upward. Even so, it was clearly a meaningful task within the company. The Google Brain team joined forces with the speech recognition team and used deep learning to noticeably raise the accuracy of Google's speech recognition.

The success of this collaboration became the occasion for other teams inside Google to come to trust Google Brain and deep learning. One success tends to invite the next. Soon after, Google Brain took on the Google Maps team, which was seeking to improve the quality of its map data, as its second internal customer. Buoyed by momentum from two consecutive successes, Andrew Ng was finally able to hold in-depth conversations with the advertising team, the company's core revenue source. An experiment that had begun in a corner of a laboratory had, within a few years, walked its way into the heart of the company.

This approach, starting from a modestly sized but clearly meaningful project and steadily building results and trust to transform an entire giant tech company into an AI company, did not end as a single success story. This was the root of the first strategy Andrew Ng would later write in the AI Transformation Playbook at Landing AI, the principle of running pilot projects to gain momentum. This method, learned firsthand at Google Brain, remained a principle he would repeatedly bring up whenever he later spoke to various companies about adopting AI.

The grammar of scaling that Andrew Ng planted at Google Brain survived beyond that moment. This team would later invent the Transformer, the most broadly scalable neural network architecture in history, laying the foundation for the generative AI wave that is transforming the world today. This approach, growing the size of models and pouring data into them so they learn to mimic the patterns hidden within that data, has since established itself as a single law running through the entire AI industry.

And yet Andrew Ng himself is today one of the figures most firmly cautioning against the notion that AGI, artificial general intelligence, will arrive within just a few years. He keeps his distance from the words of various companies that emphasize the imminence of AGI for the sake of publicity or fundraising. Even though he is someone who has believed in the power of scaling longer than almost anyone, he points out that because today's cutting-edge models have already read nearly all the text scattered across the internet, the old approach of simply scaling up is running into limits. His diagnosis is that overcoming that limit would require bearing exponentially rising costs.

As he sees it, reaching true AGI still requires several more decades. His judgment is that scaling laws alone, simply enlarging today's Transformer architecture, cannot reach genuine intelligence. He says that beyond endlessly growing models, agentic workflows that embed structured human knowledge into systems must also be put in place, and new technical breakthroughs different from what exists today are needed as well.

It is an unexpectedly measured diagnosis for someone who opened the era of large-scale deep learning. The very person who once ran 1,000 computers for three days to find a single cat in a YouTube video is now saying there is a wall that cannot be crossed no matter how many computers are added. That the person who believed in scale is now the first to point out the limits of scale suggests that this clear-eyed sobriety may be another lesson he learned at Google Brain.

To fully understand this Google Brain experiment, it helps to first look at how AI research handled data around that time. Computer vision research around 2010 was dominated by supervised learning, carried out on datasets where humans had directly attached labels to each individual image. A person would mark in advance whether a photo contained a cat, a car, or a person, and the neural network would learn to guess correctly by treating that mark as the answer. Building a dataset covering as many categories as ImageNet required that many human hands, and labeling image after image was a task that consumed a great deal of time and money. Even when trying to grow the size of a dataset, there was always a clear ceiling in the form of how many labels people could attach by hand.

The question posed by Google Brain's YouTube Cat experiment shook this very premise. It asked what would happen if a neural network were shown nothing but a pile of video images randomly pulled from YouTube, with no human-attached labels at all. Having no labels means there is no hand at all telling the network this image is a cat or this image is a person. The neural network had to establish its own criteria for classifying the world, using only the patterns that repeatedly appeared within the images as its clue. This approach is called unsupervised learning. Whether unsupervised learning could actually produce useful results was an unverified question up to that point, and Google Brain answered it with 1,000 computers and three days.

The term grandmother cell is also worth revisiting in this context. In neuroscience, there has long been debate over whether a single neuron in the brain responds only to one specific object, or whether many neurons work together to represent a single concept. If a neuron really does fire every time one sees one's grandmother's face, it would mean the brain stores each of the world's complex concepts in a remarkably efficient way. The Google Brain team's attempt to transplant this concept directly into an artificial neural network was an effort to pose neuroscience's old question anew in the language of computer science. It was an attempt to check whether this efficient way of representing concepts, as shown by the brain, could be reproduced inside an artificial neural network, even without directly mimicking the human brain.

The word sparse attached to the name sparse autoencoder is also worth unpacking. In the process where an autoencoder compresses input and then reconstructs it, this sparsity is a constraint that suppresses most of the internal neurons to stay quiet regardless of what image

comes in, allowing only a small fraction of neurons to respond. If every neuron were left free to respond noisily, the network would end up memorizing even the tiniest noise in an image, but by suppressing the number of responding neurons, the network is forced to focus only on a handful of truly important features. The term local connection refers to a design in which each neuron in the network looks only at a small region nearby, rather than the entire image. This reflects the idea that, just as a person can guess whether an eye or a nose is present by looking at only one corner of a photo, individual neurons do not need to see the entire image. Thanks to this local connection, the researchers were able to reach a scale of one billion parameters while greatly reducing the number of connections. Pooling is a device that bundles the responses of several neurons together so that the network gives the same answer even if the image shifts or shakes slightly, and local contrast normalization is a device that keeps the network stable even when an image's brightness or contrast varies depending on the shooting conditions.

The nature of the computers used in this experiment is also worth noting. When people think of training a large-scale neural network today, the image that usually comes to mind first is mobilizing a large number of graphics processing units, or GPUs. But Google Brain's early experiment ran not on graphics processing units but on a cluster made up of ordinary central processing units, or CPUs. This is one reason the numbers, 1,000 computers and 16,000 cores, had to be so large. To make up with central processing units for the amount of parallel computation a single graphics processing unit could handle, that many more machines had to be mobilized to fill the gap in computing power. It would still take a few more years before using graphics processing units in earnest for deep learning training became the industry standard. Google Brain's experiment was a case that solved the problem of scale, right before that shift took place, by pushing through with sheer numbers of machines.

Unpacking a bit further the parallelization method that DistBelief implemented is as follows. Loading an entire massive neural network onto a single computer and training it there was simply impossible from the outset. The research team cut the neural network into multiple pieces and distributed them across 169 machines, and these divided pieces had to constantly exchange information with each other, moving as if they were a single body. On top of this, the team made five replicas of the entire divided network, having each replica learn from a different piece of data at the same time. What these five replicas learned was continuously carried over to 256 parameter servers and merged into one. The problem was that this process did not run in perfect lockstep. Some replicas finished their calculations quickly, while others finished a little later. The reason they chose an asynchronous method, accepting computation results in whatever order they arrived, was that if they waited for every machine to finish at exactly the same speed, the entire training speed would be held hostage by the single slowest machine. This asynchronous processing method has since become a design commonly seen across many labs working with large-scale neural networks.

Actually running an experiment of this scale required considerable cost. Tying up 1,000 computers for three whole days was not a budget an ordinary university lab could bear. The backdrop that allowed Andrew Ng, while still at a university, to conceive of an experiment on this scale was the vast data center resources of a company like Google. One reason academic neural network research had long remained stuck on small problems that could be run on a handful of computers was precisely this gap in resources. University labs had no good way to secure computing resources on this scale, but companies that already had massive data centers to run search, advertising, or social networking services were in a different situation. Andrew Ng's choice to leave academia and move into industry can be seen not as a matter of receiving more research funding, but as a choice to move into an environment where he could actually carry out experiments at a scale that academia could not even attempt at the time.

The significance of the collaboration with the speech recognition team is also worth examining a bit further. Speech recognition is the technology that converts a person's voice into text. Behind the process of speaking into a smartphone and having those words turn into text on the screen lies this technology. When this technology's accuracy is low, users face, day after day, sentences appearing on screen that differ from what they actually said. Conversely, when accuracy rises, users do not notice the change at all and simply enjoy the added convenience. The fact that Google Brain achieved this quiet improvement was an event that quietly proved within the company, without any flashy announcement or new product launch, that deep learning could raise the quality of an actual service. It was likely not a single paper out of a lab but the fact that the everyday experience of countless users had actually changed that moved the other teams inside Google.

Reading Andrew Ng's diagnosis today about the limits of scaling against this background makes its texture a little clearer. During his Google Brain years, the limits he faced were the number of computers and the speed of gathering data. Problems got solved by mobilizing more computers and pulling in more images from YouTube. But the limit he points to now is different in character. The text that already exists on the internet is finite. If models have already read nearly all of that text, then no matter how many more computers are added, there is no new text left to read. During his Google Brain years there was YouTube, a vast new warehouse of data, but his diagnosis now is that no such easy new warehouse remains. Scaling itself is still a valid method, but what differs is that the material available to scale is no longer as abundant as before.

The term agentic workflow is also worth unpacking. It refers to a flow in which, rather than a model producing an answer to a question all at once, a single task is broken into multiple stages, and the model plans, executes according to that plan, checks its own output again, and revises it if necessary. This resembles the way a person solving a difficult problem writes a draft rather than jumping straight to the answer, then reads that draft again and corrects the

mistakes. The reason Andrew Ng discusses this approach alongside the older approach of simply growing model size is that he sees intelligence as something that cannot be completed through a single massive computation alone, but that only reaches a trustworthy result through multiple stages of checking and revision.

Looking back, what a single team called Google Brain left behind in the world does not stop at the story of one neural network that recognized a cat. The result of this experiment, that neural networks can learn the world's concepts on their own if the number of computers is increased and data is poured in, was a preview that foreshadowed the working method of the entire AI industry for the decade that followed. The sequence that labs training today's massive language models now take for granted, gathering as many computers as possible, securing as much data as possible, and then leaving the network to find patterns on its own, first took shape in a Google lab in 2011. The fact that Andrew Ng was the one who opened the door of that lab gives different weight to what he says today when he speaks about the limits of scaling.

Behind the name Google Brain there was not only Andrew Ng. Among the researchers who designed and carried out the YouTube Cat experiment was a name, Quoc V. Le. This researcher, who would later build an independent reputation in the field of neural architecture search, was at the time a core member of Google Brain in charge of the experiment's detailed design. While Andrew Ng took on the role of setting the team's direction and persuading the company internally, building the nine-layer autoencoder, running the code across 1,000 computers, and checking the results one by one was a burden shared among several researchers. This achievement was, from the very start, an achievement of the whole team, not one person.

The term gradient descent is also worth touching on. Training a neural network ultimately means gradually revising the numbers, or weights, assigned to each of countless connections. When the network's answer is wrong, its error is calculated, and the weights are shifted very slightly in the direction that reduces that error. Repeating this process millions of times, the network comes to give increasingly accurate answers. The name gradient descent comes from its resemblance to descending a foggy mountain step by step toward lower ground, feeling only the slope underfoot. The word stochastic is attached because, instead of examining the entire dataset to decide direction, the method randomly samples only a portion of the data to estimate direction. Since checking the entire dataset every single time would take far too long, moving frequently in small steps based on an estimated direction actually reaches the destination faster. The word asynchronous meant letting multiple computers continue this estimating and stepping at their own pace, without waiting for one another.

The terms fine-tuning and logistic classifier that appeared in the ImageNet experiment are also worth unpacking. Fine-tuning refers to the work of slightly adjusting a neural network that

has already completed one round of training to suit a new task, building on what it has already learned, rather than retraining the whole network from scratch. When transferring a network that had learned the concepts of faces and cats from YouTube videos to the new task of recognizing ImageNet's 20,000 categories, the researchers chose to add only the sense needed for the new task on top of the knowledge already acquired, rather than teaching it all over again from the beginning. The logistic classifier is the device at the final stage that takes the features extracted by the network and makes the final judgment about which category an image belongs to. Most of the neural network takes on the role of extracting meaningful features from an image, and this small classifier sitting on top attaches a label based on those features.

The decision to choose the speech recognition team, rather than the advertising team, as the first internal customer also carried an organizational sense. Pushing a new technology straight into a core department that carries the company's revenue and having it fail risked that failure being reflected back as distrust toward the entire technology of neural networks. By contrast, achieving results first in a department that was meaningful within the company but not directly tied to revenue meant the burden of failure was not heavy, and success would provide grounds for drawing in the next collaboration. Andrew Ng learned firsthand within Google that trust is not gained all at once, but that only after piling up several small successes does the door to a major department finally open.

The skepticism surrounding neural networks at the time also had its own background. AI research had a history of going through two so-called winters, in the 1970s and the 1980s. Each time, neural networks failed to deliver results matching expectations, and research funding and interest dried up. Even into the late 2000s, in fields like computer vision and speech recognition, methods such as support vector machines or decision trees were considered to produce more stable and predictable results, and neural networks were often treated as a technology theoretically interesting but difficult to handle in practice. Researchers who knew this history tended to be more cautious about betting heavily on neural networks again. One reason Google Brain's results had such a strong impact both inside and outside academia was that they showed, with actual data and actual numbers, that a technology which had disappointed people several times before was different this time.

By this point, Andrew Ng had already achieved results that could stand as a paper bearing his own name, and yet he put far more effort into taking root within the company than into that achievement itself. Proving a technology and getting an entire organization to adopt that technology demand different kinds of effort. At Google Brain, he carried out both of these efforts together, and this experience became the prototype for a challenge he would repeatedly face afterward, moving between academia and industry: the problem of getting people to actually use new technology.

Three Years at Baidu

In May 2014, a new option appeared before the scholar who had built his name moving between Stanford lecture halls and Coursera's servers. Baidu, the company leading China's internet search market, announced that it was opening a U.S. research and development base in Sunnyvale, California, and naming him Chief Scientist. The news drew attention from both Silicon Valley and academia. That the man who had founded and led Google Brain was now moving to head the research organization of a Chinese company rather than an American one was itself unusual.

In making this decision, he stepped down from his post as co-CEO of Coursera, though he kept the title of chairman of the board. He had not fully severed ties with the online education platform he had founded, but rather moved back from the front line of operations into a supervisory role. The private reasons why he chose a Chinese company over his stable footing in Silicon Valley and the executive position he himself had built at Coursera cannot be confirmed from the surviving record. The facts covered in this chapter are reconstructed from Baidu's opening announcement, the piece he wrote himself upon resigning, and remarks he made in later broadcast interviews. Since no internal management metrics or records of closed-door meetings survive, this chapter does not address his personal motivations or whatever cultural experiences he may have had living in China.

Robin Li, Baidu's chairman, introduced his joining by calling him a person of true vision and a key contributor to the field of artificial intelligence, saying that in an era where AI's role was only growing, Baidu had found the right person to lead its research. This assessment did not end as a simple pleasantry. Andrew Ng would later recall Robin Li as one of the few major-company CEOs who recognized the value of deep learning early on, and as still one of the world's foremost AI executives. The relationship between the two appears to have grown beyond a simple employment contract into a collaboration built on mutual trust in each other's technical judgment.

Baidu Research, which he came to lead, was reorganized during this period into three labs. The newly opened Silicon Valley AI Lab, the Beijing Deep Learning Lab which had already been pursuing deep learning research since 2013, and the Beijing Big Data Lab were brought together under a single structure. Upon taking the post, he described Baidu as a company with a long-term vision and deep commitment, saying he was excited to take part in advancing foundational technology capable of changing the world. Whatever personal ambition or academic calculation lay behind this statement is not recorded. But the judgments he made during this period can be confirmed by his actual actions over the following years, which followed through on them.

He has stated that from 2014 to 2017, during his time at Baidu, he stood at the head of the roughly 1,300-person Baidu AI Group as a whole, which included the 300-person Baidu Research. This figure comes from his own recollection rather than from the company's official statistics, so it is difficult to verify the precise organizational chart or staffing. Still, various circumstances support the idea that the scale itself was not small. How he actually managed an organization of this size, what evaluation systems he put in place, and what friction arose in transplanting Silicon Valley-style research culture into a China-based organization cannot be confirmed from the surviving record. It is more honest to leave this as a gap in the record.

Even so, the piece he wrote upon resigning names several colleagues who led the organization alongside him. He worked closely with Chief Operating Officer Lu Qi, an executive with deep AI experience, and built the organization together with Haifeng Wang, a researcher and technical leader, and Yuanqing Lin, who had capabilities spanning both technology and business. Beyond these, he recalled discovering and working with a number of other researchers, including Adam Coates, and devoting himself to building the organization's intellectual foundation. The impression left is less of an organization gathered under one name than of a structure in which several leaders divided the responsibilities among themselves.

One experience he valued for himself during this period was the chance to witness the AI ecosystems of the United States and China at the same time. He assessed that while the United States was strong at generating new technological ideas, China was strong at turning those ideas into products and bringing them to market quickly. He regarded it as a valuable experience to have had a foothold in both technology ecosystems, each with its own distinct strength. Since there is no documentary support for exactly what concrete changes this observation left on his later research or management approach, this account passes along only the remarks he left, as they stand.

What the Baidu AI Group did was not limited to writing papers and publishing research results. He stated that the software his team built was used daily by hundreds of millions of users in practice. The team's work was broadly divided into two directions. One was supporting Baidu's already-established core businesses. Even before he joined, Baidu was already ahead in image recognition, speech recognition, natural language processing, and machine translation, and his team built on this technical strength to carry out dozens of projects across nearly every business area of the company, from search and advertising to maps, food delivery, voice search, security, and consumer finance. He referred to the existing businesses using the term *mothership*. This meant the work was closer to refining the course and increasing the speed of a large ship already at sea than to creating new technology.

The other direction was building entirely new businesses from scratch. He stated that in the final two years of his tenure, he launched a new independent business unit into the world

every year. He personally took charge of the autonomous vehicle team and set it on track, and he led a smart-speaker business built on speech recognition technology, establishing a conversational computing platform capable of competing with similar products from Amazon and Apple. He later recalled with pride that these teams were still producing results in the Chinese market. Beyond these, new efforts also took root during this period, including facial recognition technology applied to ticket gates that automatically open as an authorized person approaches, and Melody, a conversational AI for the medical field. He referred to himself as the architect of Baidu's AI strategy, and this phrase carries a sense of pride in having led the process by which AI took root within the company.

The clearest legacy left by these three years is that he learned firsthand, through direct experience, how to integrate AI into business within a large organization. Where his experience at Google Brain had been building a research organization, at Baidu he directed, from start to finish, the process of turning that research into actual revenue and products. This experience later became the foundation for the guides he wrote to help various companies with their AI transformation after he founded Landing AI. His sense of how to introduce AI within a company, what organizational structure should support it, and in what order a business should be expanded appears to have originated in the practical experience of this period.

In his third year, he announced that he was stepping down from his position leading the Baidu AI Group. Sources differ by a few days on the exact timing. The Chinese Wikipedia records the date as March 20, while the resignation piece he wrote himself was made public on March 22. Whether this gap of a few days reflects the difference between the announcement date and the public release date, or some other administrative circumstance, cannot be confirmed from the surviving record. However, judging that the date of the piece he wrote and released to the world himself carries the most credibility, this chapter takes the date his statement was made public as the reference point for his resignation. After him, Haifeng Wang took over the Baidu AI Group and Yuanqing Lin took over Baidu Research, each carrying the organization forward.

He once explained directly, in an interview at the World Economic Forum, why he left a position of such authority in a stable organization. Looking at the org chart, he said, he came to feel that the team would do well even without him in that seat. A team already filled with excellent people was firmly supporting the existing businesses, but he recalled that the moments he himself had been most absorbed in were the ones spent building a new business from nothing, as with autonomous driving or the smart speaker. Continuing such efforts within a large company was meaningful in its own right, but his conviction that he could put his abilities to better use by building an entirely new business from the ground up, in the form of a venture studio, remains the clearest motive behind his resignation. Other reasons that

outsiders might find interesting to speculate about — Chinese government censorship issues, conflict with management, personal family matters — are not confirmed anywhere in the record he left behind. Passing along such rumors as if they were fact would be an imprudent way to write this account.

The title of the piece he wrote upon resigning carried the meaning of opening a new chapter in my AI journey. In this piece, he once again stated his core belief: that AI is the new electricity. Just as electricity had reshaped the face of numerous industries over nearly a century, his outlook was that AI, too, would transform the shape of nearly every major industry, including healthcare, transportation, entertainment, and manufacturing, and improve the lives of countless people. Just as the earlier industrial revolution had freed people from repetitive physical labor, he hoped that AI would now free people from repetitive mental toil, such as gripping a steering wheel on a congested road.

He judged that he had done his part in turning two major companies, Google Brain and Baidu, into genuine AI companies. But he concluded that the possibilities AI held extended beyond the business scope of a handful of giant companies. After leaving Baidu, he put this judgment into action. He opened a new phase by founding, one after another, the venture studio AI Fund, the enterprise AI company Landing AI, and the education institution DeepLearning.AI, working to extend AI technology across traditional industries broadly and to let more people learn AI.

The three years he spent at Baidu were not, in his career, merely a period of growing one company's technology. It was a time in which he mastered the method of turning research into an actual business within an organization, and also a time in which he strengthened his resolve to carry that method to more companies in the wider world. His choice to leave a large, stable position and again build a business from the ground floor appears to be an extension of the pattern of judgment he had repeated in moving back and forth between academia and industry. His path — carrying ideas validated in the lab into the classroom, then carrying the hunger he encountered in the classroom back out into industry — was, with this declaration made upon leaving Baidu, already moving toward its next post.

Even after he left Baidu, the way he had built the Silicon Valley AI Lab in Sunnyvale remained the prototype for several organizations he built afterward. Instead of opening a new research base in the United States, he chose the approach of absorbing two labs already established in Beijing under a single chain of command. Rather than building an entirely new organization from scratch, his hand went first to gathering scattered capabilities together and setting a shared direction. This approach differed in character from his experience at Google Brain, where he had started with a small staff and grown the organization over time. At Baidu, he began by integrating the two Beijing labs that already existed with the newly created U.S. lab, and this integration work itself was the first task assigned to him.

One word he frequently invoked during this period was scale. Since his time at Google Brain, he had demonstrated through experiment his conviction that increasing the size of a neural network raised its performance in turn, and at Baidu he gained the opportunity to test this conviction again, this time atop the massive service that is search. Applying a new model to a service used daily by hundreds of millions of people demands a different kind of responsibility than improving accuracy figures in a laboratory. If a service slows even briefly or malfunctions, the impact reaches users immediately. Behind his team's ability to run dozens of projects at once across search, advertising, maps, and food delivery services must have lain the organization's accumulated experience, refined through repeatedly putting research results onto live services. However, exactly what verification steps this process went through, and how failed projects were handled, cannot be confirmed from the surviving record.

His recollection that he launched the autonomous vehicle business and the smart-speaker business, one new one each year, gives some sense of what role he played within the organization. Managing an existing business and building a business unit from nothing require different abilities. An existing business can simply follow already-set goals and metrics, but a new business must decide from scratch what to build, what kind of organization to start with, and by what standard to judge success. The fact that he launched a new business like this every year in the latter part of his tenure suggests he may have gradually devoted more of his time within the organization to designing new businesses rather than to repetitive management work. His later statement that founding new businesses in the form of a venture studio suits him better than doing so within a large corporation does not seem unrelated to the inclination he discovered in himself during this period.

Another role he took on within Baidu was finding talent. In the piece he wrote upon resigning, he called out, one by one, the names of the many researchers and executives he had worked with. Letters that list colleagues' names so specifically in this way are uncommon. It leaves the impression that he tried to describe his three years leading the organization not merely as his own achievement, but as a result built together by many people. Among them, he credited the researchers he had discovered with helping to build the organization's intellectual foundation, a pattern he has repeated whenever he has built a research organization. Whether founding Google Brain or, later, DeepLearning.AI or Landing AI, he chose to find people he could trust and divide up roles rather than handling everything himself. His experience at Baidu can be seen as a place where he tested this approach at a much larger scale.

His observation comparing the U.S. and Chinese AI ecosystems side by side was not a simple, easy impression; it left a trace in the direction of several organizations he founded afterward. He assessed that the United States was strong at generating new technological ideas, while China was strong at turning those ideas into products and bringing them to

market quickly. Turning this observation around, it also implies that the speed of moving an idea into an actual product is what determines the value of a technology. That Landing AI, which he founded after leaving Baidu, devoted itself to guiding companies through the concrete procedures and sequence of adopting AI may be connected to having directly experienced at Baidu how difficult it is to close the distance between idea and product. However, since he has never stated this causal link himself, it is noted here only as a circumstantial connection.

The structure by which Baidu Research was reorganized into three labs foreshadows a principle he would repeat when building organizations afterward: placing research staff scattered across different regions under a single chain of command, while clearly separating the domain each lab is responsible for. It can be inferred that the Silicon Valley AI Lab took on the role of exploring new research directions, while the two Beijing labs took on the role of turning the image, speech, and natural language processing technology they had already accumulated into actual services. This division of roles is not documented explicitly, but looking at the character of the projects he led, there is evidence that research and application were also to some degree separated geographically.

The three-year span itself is also worth noting. His time leading Google Brain was not long, nor was his time as Coursera's co-CEO. The three years he spent at Baidu likewise sit on an extension of a pattern: rather than staying long in one organization, he moves to the next post once he has solved a particular problem. Whether he designed this pattern deliberately or simply followed along naturally each time a new opportunity arose cannot be determined with certainty. Looking at the results, however, he has built his career by mastering a method of solving problems within one organization and then carrying that method to the next. Baidu was, within this flow, the place where he personally ran the largest organization, and this experience became the practical foundation that later allowed him to stand in a position advising various companies on their AI transformation.

Even after the May 2014 announcement, various interpretations circulated within and beyond Silicon Valley over why Baidu specifically. A case in which a company that had contended with Google in the search engine market — listed on U.S. stock exchanges but with its headquarters and revenue base in China — placed an American scholar at the head of its AI research had, up to that point, been uncommon. Silicon Valley media read this appointment as a signal of Chinese companies drawing in talent from American academia, while academic circles took note of his unusual dual arrangement of keeping his Stanford professorship on a concurrent basis while heading a large company's research organization. However, what weight this outside perspective carried in his own decision does not appear in any known record.

The Silicon Valley AI Lab that opened in Sunnyvale had a different character, even down to its physical space, from the two Beijing labs. With the center of gravity of the existing search and advertising business located in Beijing, rather than building a new base on American soil, he chose the path of drawing the research capability already accumulated in Beijing under his own command. This structure demanded a form of organizational management he had not encountered before. At Google Brain he had started with a small staff and built up the organization achievement by achievement, but at Baidu, the first task given to him from the outset was integrating two already-established labs with the newly created U.S. lab. This means that the work of his first hand was aligning geographically separated organizations toward a single direction.

What he had to manage during this period was not research staff alone. Layering new technology onto services already used by hundreds of millions of people — search, advertising, maps, food delivery — demanded an entirely different kind of responsibility than research improving accuracy figures in a laboratory. If a service slowed even briefly or produced an error, the impact reached users immediately. Experimenting with scaling up neural networks and raising accuracy atop the massive service that is search carried a different weight than the scaling work he had done within the lab during his Google Brain years. Behind his team's ability to run dozens of projects at once across such a broad range of areas — search, advertising, maps, food delivery, voice search, security, consumer finance — is presumed to lie an accumulation of organizational experience, refined through repeatedly putting research results onto live services. However, exactly what steps this process went through in detail, and by what standard failed attempts were wrapped up, cannot be confirmed from the surviving record.

Sustaining an existing business and building a new one from scratch require different capabilities from one another. An existing business can follow already-set goals and metrics while continuing to improve, but a new business must decide anew, each time, what to build, what organizational size to start with, and by what standard to judge results. His recollection that he launched a new business unit into the world every year during the final two years of his tenure suggests that during this period he may have devoted more and more of his time to the role of designing new businesses rather than that of an organizational manager. The experience of setting the autonomous vehicle team and the smart-speaker team on track, and of incubating facial recognition technology and a medical conversational AI, are all concentrated around this time. He confirmed firsthand that such efforts were possible even within the large structure of a major corporation, but at the same time, he also appears to have come to hold the question of whether this approach was truly best for him.

The three years he led Baidu stand out even when set against the flow of his earlier career. His time leading Google Brain and his time as Coursera's co-CEO had likewise not been long.

He built his career in a pattern of mastering a method for solving a particular problem within one organization and then carrying that method over to apply it in the next. Baidu was, within this flow, the place where he personally ran the largest organization. The experience of overseeing a staff of some 1,300 people while simultaneously tending to the core business of search and the fledgling business of autonomous driving appears woven into the practical sense behind the advice he later gave companies in founding Landing AI. This three-year shift — from a scholar leading a handful of researchers to an executive running an organization of thousands, responsible at once for both revenue and product — appears to be distilled into practical instinct in the advice he later offered companies.

His observation of having experienced the AI ecosystems of the United States and China side by side also originates in the experience of this period. His diagnosis that the power to generate new ideas and the speed of turning those ideas into products grew up in different countries also implies that closing the distance between idea and product is what determines the value of a technology. Behind Landing AI's, founded after he left Baidu, devotion to guiding companies through the concrete procedures and sequence of adopting AI may lie his own direct experience at Baidu of just how difficult it is to close that distance. However, since he has never stated this causal link himself, it is noted here only as a circumstantial connection.

The way he called out his colleagues' names one by one in the piece announcing his resignation is also worth noting. A resignation letter that lists the names of people worked with so specifically in this way is not a common format. It leaves the impression that he tried to describe the achievements of these three years not merely as his own, but as a result built together by many people. This attitude recurs in the organizations he built afterward as well. Whether founding Google Brain or, later, DeepLearning.AI or Landing AI, he has consistently chosen to find trustworthy colleagues and divide up roles rather than take on every role himself. His experience at Baidu can be seen as a place where he tested this way of running an organization at a much larger scale.

By the time he left Baidu, he appears to have already sketched out his next step to some degree. His recollection of looking at the org chart and gaining the conviction that the team would do well without him reads not as merely the venting of a tired executive but as a judgment aimed at the next stage. Through the experience of launching new businesses one after another within the large ship of a major corporation, he confirmed that the moment he became most absorbed was when building something from nothing, in a place with nothing there before. This confirmation led directly to his choice, immediately after leaving Baidu, to found an organization in the form of a venture studio. Between experimenting with new businesses using the resources and scale of a large corporation and building a business as an independent organization from the start, he chose the latter, and underlying this choice are

both of the experiences he had directly at Baidu: managing a massive organization, and growing a fledgling business within it.

The date Baidu's U.S. research and development center opened was May 16, 2014. The press release issued that day names the organization he would lead specifically. Baidu called its entire artificial intelligence initiative Baidu Brain, and he took the seat overseeing this initiative. That a company running a search engine attached the name brain to its core technology strategy reads as meaning the company regarded AI not as a peripheral feature but as something like the nervous system supporting the entire company. Whether this name was already in use before he joined, or was newly settled upon in connection with his joining, cannot be confirmed from the surviving record.

Looking at the composition of the staff in the organization he led reveals that it was not simply a team gathered from researchers alone. In the piece he wrote upon resigning, he wrote that his team was thickly filled with talent across multiple layers, from executives to outstanding engineers, scientists, and product managers. Running a single research lab and running an organization that spans the entire process from research to product launch require different staff compositions altogether. People who only write papers cannot put technology onto a service used daily by hundreds of millions of people, and people who only build products cannot set the direction of that technology. Behind his ability to build an organization capable of handling both roles at once appears to lie an eye for hiring and placing researchers and product staff side by side.

The people named in the resignation letter were not limited to Lu Qi, Haifeng Wang, Yuanqing Lin, and Adam Coates. He also named researchers such as Jing Kun, Liping, Xu Wei, and Zhu Kaihua, writing that they too had contributed to building the intellectual foundation of Baidu's AI organization. Cases of naming as many as eight people together in a single resignation piece are uncommon. A final piece written on the way out of an organization is generally expected to use its space to sum up one's own accomplishments, yet he devoted a considerable portion of that space to calling out his colleagues' names. This approach leaves the impression that he tried to portray himself not as the person who built the organization, but as a person who had passed through it.

The scale of the organization he handled during this period was incomparable to his Stanford classroom or the early days of Google Brain. Unlike the years he spent writing reinforcement learning papers with a handful of graduate students, or the early days of Google Brain when it started with just a few staff, at Baidu he inherited, from the outset, a staff of over a thousand people and an already-established business structure. Taking on an organization of this scale means spending more time coordinating priorities among multiple teams and allocating resources between ongoing businesses and newly starting ones, rather than on setting research direction. It can be surmised that the balance of prioritizing resources

for revenue-supporting businesses like search and advertising while also assigning staff to businesses not yet generating revenue, like autonomous driving and smart speakers, must have been a practical challenge he faced every day.

This experience stands as a clear turning point compared to the path he had walked before. What he built up through Carnegie Mellon, MIT, and Berkeley was academic achievement, and what he built up as a Stanford professor was the experience of training students. At Google Brain, he achieved the experimental feat of being the first to prove that large-scale neural networks actually work. Baidu layered onto all of this experience the managerial challenge of pushing new technology into the business structure of an already-established large corporation while simultaneously growing new businesses. It can be said that these three years played a decisive role in his transformation from a researcher into someone equipped with the instincts of an organizational executive.

Looking at the various organizations he built after leaving Baidu reveals that he made choices running in a different direction from the scale-related challenges he faced during this period. AI Fund was designed from the start not as one large organization but as a structure supporting multiple small founding teams, and Landing AI set out as an advisory-natured organization helping multiple companies adopt AI, beyond any single company. He turned away from the experience of managing a staff of over a thousand and back toward tending to several small-unit teams at once. The fact that someone who had run a large organization then chose small units again suggests that the conclusion he drew from Baidu may have been less about how to grow scale than an insight into what is lost once scale grows large. His recollection of gaining the conviction that the team would do well even without his seat on the org chart appears to have led to the judgment that a role tending to several small beginnings at once suited him better than that of a large organization's manager.

Coursera and Lowering the Threshold for Online Education

In 2011, a single classroom on the Stanford campus was quietly changing the world. That year, Andrew Ng opened Stanford University's machine learning course online, and enrollment surpassed 100,000 students—a scale far beyond what any number of students could ever sit face to face on a university campus. This course remained one of the early massive open online courses led by Stanford and was recorded as one of the largest online classes in the world. The goal he held when he began this work was clear: to let anyone, anywhere in the world, receive a good education for free. Why a professor at an elite university poured such effort into education without walls, and what personal conflicts or formative experiences may have lain behind that resolve, cannot be confirmed from available records. Still, the goal he set and the method he chose to realize it become clear through the path he took afterward.

The online course that drew in 100,000 students did not end as a one-semester experiment. The following year, in 2012, Andrew Ng founded Coursera together with Daphne Koller. The idea was to deliver university-level lectures online at massive scale. There is no way to confirm from surviving records what process the two went through in deciding to become partners, or what discussions or disagreements may have passed between them. Judging by the outcome, however, Coursera's launch sent no small ripple through the education world that year. The New York Times called 2012 the year of the massive open online course and gave the trend concentrated coverage. Leading universities and companies around the world partnered with Coursera, and courses, professional certificate programs, and degree programs were added one after another. Over time, Coursera established itself as the largest massive open online course platform in the world, with enrolled learners surpassing 150 million. Andrew Ng served as Coursera's co-CEO until 2014. That year he moved to Baidu, but his ties to Coursera were not severed—he remained on as chairman of the board and co-founder, continuing his relationship with the company.

No less important than building the vessel that was Coursera was the work of refining the courses placed inside it. The course that has followed Andrew Ng's name the longest is the machine learning course. First opened in 2012, this course recorded a 4.9 out of 5 student rating and has been taken by more than 4.8 million learners. As the years passed, the primary programming language of the AI industry shifted from Octave to Python, and in step with this shift Andrew Ng rebuilt the old course into a three-course Machine Learning Specialization. At the center of the redesign was a single question: how could people unfamiliar with math and coding be brought along to follow the material? So the new program chose an order that showed concepts first through pictures and graphs, then had learners write out the algorithms in code—a method that let people get a feel for things by eye first, then confirm it by hand,

rather than throwing equations at them and demanding understanding. Joining this redesign work were people such as Eddy Shyu, Aarti Bagul, and Geoff Ladwig. Aarti Bagul was a machine learning engineer who had worked with Andrew Ng at both Stanford and Landing AI, and Eddy Shyu was a product leader who had built multiple educational programs at DeepLearning.AI. Records do not show how these people came to form a team, but the newly designed Machine Learning Specialization still has more than 812,608 learners enrolled on Coursera today.

The mark a single course leaves on a person's life cannot be fully explained by numbers alone. Chirag Godawat, a data engineer, became a machine learning mentor while taking this program, published a research paper with IEEE, and ultimately landed a position at JPMorgan Chase. Nicholas Muchinguri, a senior trader at Vanguard Australia, used what he learned in this course to automate his investment process. Akash Saroop, then a student, wrote a paper on facial expression recognition and used it to land a Morgan Stanley internship. A particularly notable case is Nektarios Kalogridis. Unemployed at the time, he looked back on taking this machine learning course as the biggest turning point of his life, and went on to found a startup combining deep learning with Wall Street data. It would be hard to claim these cases represent the millions who have passed through this course as a whole. Even so, these few stories concretely show that a single online classroom can open a new path without a degree or connections.

Once the machine learning course was established, Andrew Ng chose a path of broadening education at both ends at once. At one end he placed a deeper, more specialized program. Partnering with Kian Katanforoosh, a Stanford Computer Science lecturer, and Younes Bensouda Mourri, he opened the Deep Learning Specialization. Made up of five courses, this program targeted intermediate learners and demanded a substantial commitment of ten hours a week over three months. It was not an easy path, but it served as a practical route in for those trying to break into the AI industry, and more than 993,144 learners have enrolled in this program to date.

At the other end, he made a place for people who had never written a single line of code. This was the course called AI For Everyone. It targeted managers who work at companies rather than programmers or engineers, people who deal with policy, and the general public simply curious about what artificial intelligence is. More than 2,563,631 people have taken this course to date. It is structured as four modules totaling seven hours. The first week covers what AI is as a concept, the second week covers how to set up an AI project within a company, the third week looks at bringing AI into an organization as a whole, and the fourth week addresses AI's impact on society. At the center of this course was sorting out what terms like neural networks and deep learning actually mean, and where the line falls between what AI can do now and what it still cannot. Questions that go beyond the technology itself—biased

judgment, adversarial attacks meant to deceive people, and effects on jobs—were addressed within it as well. It was work that pulled AI, which had remained a language known only to experts, down to a place where people who did not know that language could still make judgments about it.

What Andrew Ng and Coursera tried to lower the threshold on was not only the difficulty of knowledge. It was also cost, and no small barrier at that. The guidance documents for the courses introduced above each carry a financial aid item. For learners who cannot afford the \$49 fee for a certificate or the \$30 monthly subscription, Coursera runs a financial aid program. Once an application is submitted and approved, learners can receive the same courses and certificates as paying students at no cost. Without such a mechanism, no matter how well a course was designed, only those with spare money to begin with would ever have been able to knock on its door.

There is one story Andrew Ng himself has told of how this lowering of the financial threshold actually changed a life. When he partnered with Google to launch the TensorFlow Professional Certificate program, Google's spending on the program came to about \$100,000 a year. Measured against the scale of a giant company's finances, it was not a large sum, but the trust and goodwill this small investment produced far exceeded that figure. Andrew Ng recounted meeting a young man from Syria at an event in Amsterdam. Amid a civil war that was shaking his entire country, this young man became one of the first people in Syria to earn a TensorFlow certificate. That certificate became a stepping stone out of poverty for him, and he moved to Germany and landed a position at a major company. This story is moving, but it is a single case. There is no basis for concluding, from this one case, that the same thing happened everywhere in the world. Still, this story is enough as concrete evidence that a single online certificate can, in some country, actually change the course of a life.

The current that began in a Stanford classroom in 2011 has, through Coursera and DeepLearning.AI, delivered AI education to more than 8 million people to date. There is no remaining basis for concluding that this number has reached every region of the world evenly. Data detailed enough to determine exactly how much it has reached which countries or which classes of people is lacking. Even so, what this current shows is clear: a class that began in a single classroom crossed over the walls of the university, across gaps in language and income, and reached into the concrete lives of individual students. The single online course Andrew Ng created at Stanford kept extending outward—into the company that became Coursera, into the many courses within that company, and into the personal histories of the millions who took those courses. DeepLearning.AI, the subject of the next chapter, is the point where this current extends beyond Coursera into another organization.

Looking at how the actual courses ran within the vessel that was Coursera makes clearer what lowering the threshold concretely meant. A week's worth of instruction was broken up

into several short videos of around ten minutes each, and a short check-in question followed every video. Anyone who lost the thread partway through a lecture could rewind the video on the spot, watch it again, and answer questions to confirm for themselves whether they had understood it properly. In a university classroom, there might be only one chance to raise a hand and ask a question, but online, one could go back over the material as many times as needed until it clicked. In courses dealing with code, such as machine learning or deep learning, an assignment requiring an actual program to be written and submitted came at the end of each week, and it was graded automatically with results returned immediately. Whether a student sat on the Stanford campus listening to the lecture or connected from the other side of the world with a single laptop, the grading and feedback they received were no different. Rather than having a single teaching assistant flip through grading sheets by hand and make students wait their turn, a structure was built where the machine returned answers instantly. On top of this, each course had its own discussion board, so that students stuck on the same problem could exchange questions with one another and work toward answers together. Since a single professor could not possibly attend to hundreds of thousands of people individually, a space was created for students to teach one another and fill that gap.

There were two ways to take a Coursera course. One was auditing—looking over the lecture videos and materials without paying. The other was paying and submitting assignments to receive an official certificate. Auditing alone was more than enough to gain knowledge, but it left behind nothing that could be listed on a resume or job application. To receive a certificate, by contrast, one had to pay a one-time fee of \$49 depending on the course, or a \$30 monthly subscription. The financial aid program discussed earlier was the device that erased this boundary. Learners who applied and were approved received the same lectures, the same assignments, and the same certificate as those who paid. There were, in effect, two doors—one to obtaining knowledge and another to proving it—and even the second door was kept open to those in difficult circumstances. Without this dual structure, no matter how excellent the lectures were, job seekers or workers preparing to change careers who needed a certificate would ultimately have been narrowed down to those who had money.

There was also a reason the Machine Learning Specialization and the Deep Learning Specialization were each split into several courses rather than offered as one. The Machine Learning Specialization was built from three courses and the Deep Learning Specialization from five, the result of a design that had learners climb step by step rather than throwing a vast amount of material at them all at once. It was a structure in which concepts learned in an earlier course became the premise for the next, culminating in a final project that tied everything learned together. The difficulty of each stage was linked tightly enough that even someone unfamiliar with math and coding would be ready for the second course after finishing the first. A learner who lost interest partway through, or found they no longer had time to spare for other reasons, could take just one course and still receive a self-contained

certificate for it, and if circumstances allowed, could move on to the next course and complete the whole specialization. Rather than forcing all five courses on someone as a single block from the start, the design left room for learners to divide their steps according to their own circumstances.

For a course built this way to actually carry weight in the job market, universities and companies willing to recognize its certificate had to be present alongside it. As mentioned earlier, Coursera partnered with leading universities and companies around the world and added courses along with professional certificate and degree programs. The TensorFlow Professional Certificate that Google built in partnership with Andrew Ng was also a product born out of this structure. A certificate carrying the names of a university and a company together held a different weight than completing some nameless online course. Employers gained a basis for distinguishing whether an applicant had attached material made up arbitrarily on some unfamiliar site, or had completed a program designed and graded jointly by a verified university and company. Behind the fact that the cases of Chirag Godawat, Nicholas Muchinguri, Akash Saroop, and Nektarios Kalogridis discussed earlier could actually lead to jobs, internships, and startups was this structure of recognition. No matter how excellent a course itself was, without a basis for trusting its output, it would have amounted to no more than a single line on a resume.

The weight carried by this whole current becomes clearer when set against scale. At Stanford and other elite universities, a student taking one course in one semester numbers, at most, a few hundred. Classroom size and the amount of grading a single professor can handle set that limit. The 100,000 people who flocked to Andrew Ng's online machine learning course in 2011 was a number that, in a single course, easily surpassed the classroom capacity Stanford had managed for over a century. And this current did not end as a one-time event—it carried over into the company that became Coursera and settled into a structure that has delivered AI education to more than 8 million people to date. A single online platform overtook, within little more than a decade, the number of graduates a university has produced over centuries. This comparison is not grounds for diminishing the value of university education. It is, however, a concrete figure showing how differently education can spread once the physical constraint of the classroom is removed.

Machine learning, deep learning, and AI For Everyone—the courses Andrew Ng laid out all ultimately return to a single question: how far can the barrier between a person who wants to learn and knowledge itself be lowered? For people without a math background, he showed pictures and graphs first; for people in difficult circumstances, he arranged financial aid; and to earn trust in the job market, he attached the names of universities and companies together. Refining the content of the lectures and refining the system surrounding the lectures proceeded side by side. This experience accumulated at Coursera carried over intact into the

next education organization Andrew Ng founded. The courses built under the broad umbrella of a university now stepping out from under that umbrella and establishing themselves under an independent name was the next stage of this current.

Before the name Coursera existed in the world, the online course Andrew Ng taught was called, within Stanford, by a new name: the massive open online course. Massive referred to the scale of enrollment, and open meant that there was no entrance exam, no enrollment cap, and no tuition required to get into the course. In front of the door universities had guarded for centuries, there had always stood admissions screening and tuition. The course Andrew Ng created in 2011 was a place that removed both of those gates at once. With nothing more than an internet connection and a laptop, anyone could take a Stanford professor's course, and this single condition produced the figure of 100,000. Behind the New York Times calling 2012 the year of the massive open online course was not Stanford alone. Around the same time, several other universities and educational organizations also put out large-scale online courses of the same kind one after another, and the press bundled this whole current together and reported it as a single phenomenon. Coursera, founded by Andrew Ng and Daphne Koller, was one of several attempts that emerged within this current, but over time it grew into a case ranked among the largest in the world.

The decision to found Coursera was a different decision from simply continuing to run a single online course. A school could have gathered 100,000 students by reopening the same lecture every semester. But Andrew Ng and Daphne Koller did not leave this course as an event belonging to a single university—they built a new vessel, a company, on which multiple universities and institutions could jointly place their courses. With this decision, Coursera became not a place that replayed a single Stanford course, but a place that gathered courses from universities around the world in one location. Daphne Koller, too, was a Stanford Computer Science professor who taught students from the podium, and the company the two founded together served as a bridge connecting the old institution of the university with the new channel of the online world. A scale that could not have been contained under the name of a single school was contained instead in the form of a platform in which multiple schools participated together.

The fact that Coursera did not stop at universities alone but also joined hands with companies shows the point where online education crossed from the realm of academia into the realm of industry. Traditional university lectures were built with a degree as the goal, and the body that recognized that degree was other universities and academia. The various certificate programs placed on Coursera stepped outside this frame. That companies like Google directly participated in designing certificates meant that the standard for judging those certificates was not academic recognition but the needs of the job market. Alongside the degree system universities had built up over a long time, another system was placed—one

proving practical skills companies needed, in a short period. These two systems settled into a relationship of standing side by side rather than pushing each other out. Those seeking a degree still went to universities, and those looking to quickly acquire one particular skill and leave evidence of it turned to Coursera's certificate programs.

When Andrew Ng stepped down from his position as Coursera's co-CEO in 2014 to move to Baidu, it did not mean his relationship with the company had ended. He remained in the position of chairman of the board and co-founder. He let go of day-to-day management, but kept his name attached to the direction the company had set when it was founded. Leaving management to others, he stayed in the more fundamental role of deciding what content to teach and how to teach it. In fact, the Machine Learning Specialization and Deep Learning Specialization he built afterward came into the world under the name of a course designer, not an executive. He divided the company's daily operations from the content of its courses, handing the former over to other executives while continuing to hold onto the latter himself.

A scale of 150 million enrolled learners and 8 million AI education students exceeds what any single country's university system could handle. What this scale means is that the people who passed through Coursera did not stay confined to Stanford or to the United States. Because the same lecture could be taken from anywhere with an internet connection, this education was designed, from the start, effectively without borders in mind. A channel through which technology born in Silicon Valley and raised in Silicon Valley could spread beyond Silicon Valley was opened at this point. Fields dealing with cutting-edge technology have generally tended to develop only in a handful of cities where research funding, experimental equipment, and professors who can teach the field are concentrated. The online course Andrew Ng created was a way of bypassing that geographic constraint with a single course. In regions without good universities, in countries without professors to teach AI, people could now take Stanford-level lectures with a single laptop.

As education of this scale spread, a question naturally followed: would online courses push out traditional university education? The path Coursera has actually walked shows one answer to this question. Coursera did not try to replace universities—it chose instead to join hands side by side with leading universities around the world. The lectures were held on the Coursera platform, but the places that created those lectures were still universities such as Stanford. A structure took hold in which universities provided the content and name of the lecture, while Coursera provided the channel that made the lecture viewable anywhere in the world. Within this structure, universities did not disappear. If anything, the weight carried by the name of a university reached further through this new online channel. The authority a university held only within its campus came to reach millions of people beyond that campus.

The fact that a structure allowing free lectures and a structure requiring payment for an official certificate stood side by side reveals that receiving an education and proving that

education are two different acts. Gaining knowledge required no money, but showing others that one had gained that knowledge still carried some price in some form. This dual structure is less a unique invention of Coursera's than a way of doing things that naturally took hold as online education spread at scale. Anyone could access the knowledge itself, but anyone wanting to prove that knowledge to the world had to go through some minimal procedure. This procedure was not entirely free, but because the financial aid program opened the same proof to those in difficult circumstances as well, a path was in place to obtain both knowledge and proof without money.

Looking back over this current that began in a Stanford classroom in 2011, what Andrew Ng achieved through Coursera was not limited to growing a single course to a large size. It stands as a case that showed how an old institution like the university could expand its influence upon meeting a new channel like the online world, and how companies and academia could cooperate side by side in that process. Even after he stepped back from management to remain a course designer, Coursera kept growing, and the methods forged during this period carried over intact into the organization he founded next. From an experiment that began in a single classroom to a system in which universities and companies around the world now participate, the same question always stood at the center: how far can knowledge be sent outward?

Among the programs Coursera, the company, placed in partnership with universities were not only individual courses or specializations bundling a few courses together. Over time, formal master's degree programs were added to this list as well. Unlike certificates that finish online in a few weeks, degree programs opened up, through an online path, the diploma itself that a university issues. A diploma that could once only be obtained by paying tuition and enrolling on campus could now be received, under the same name, by learners who connected over the internet. Within this structure, Coursera placed certificate programs and degree programs side by side, and learners could choose a certificate lasting a few weeks or a degree lasting several years depending on their goals. Someone preparing to change jobs with a short certificate and someone looking to build a new career with a formal degree each found their own door to walk through on the same platform.

Stacking these multiple layers of programs within a single company was also an exercise in balancing the commitment to share knowledge for free against the reality that a company must keep running. Money had to come in from somewhere for the company to stay afloat while keeping lecture videos and materials open for free. The answer Coursera chose was to price not knowledge itself but the procedure that proves that knowledge. It charged those who wanted documents bearing names such as certificate, credential, and degree, while leaving the lecture content open as-is to those who had no need for such documents. The financial aid program was the exception mechanism that issued the same document for free

to those unable to afford this cost. The company's finances and the goal of free education thus kept running together, each doing its own part in a different place.

The fact that the AI For Everyone course targeted people who were not programmers reflects, in itself, the nature of the problem this course addressed. The person who decides to bring AI into a company is usually not the engineer writing code but the manager holding the budget. Judgments about which department to bring AI into first, whether to outsource it or build a team in-house, are often left in the hands of people who do not know the technology. This is why the course's second week covered how to set up an AI project within a company, and its third week covered bringing AI into an organization as a whole. Rather than teaching code, it taught people who did not know code how to ask the questions that gauge whether an AI project will succeed or fail. Questions such as whether the data is sufficient, whether the problem is clearly defined, and whether the expected accuracy is realistic could be learned even by someone who was not an engineer, and a manager who learned these questions became able to judge for themselves which projects to push forward at their company and which to set aside.

Looking at the Machine Learning Specialization and the Deep Learning Specialization together also reveals that there was a layering between the learners each targeted. The Machine Learning Specialization was an entry point that even someone unfamiliar with math and coding could start from, while the Deep Learning Specialization was the place someone who had passed through that entry point moved on to next. Because the sequence was designed so that finishing one course naturally led into the next, learners could gauge for themselves, within Coursera, where they currently stood and what to learn next. A university would have arranged this sequence through the institutions of grade level and prerequisite courses, but Coursera made the arrangement of its courses alone serve the same function without any such institution. Beginners could come in without fear, and learners who had built up skill had somewhere further to advance into more difficult material.

Worth noting within this structure is that the people who worked with Andrew Ng in designing his courses did not stay confined to Coursera alone. Aarti Bagul also worked with him at Landing AI, and Eddy Shyu built multiple educational programs at DeepLearning.AI. The collaboration that began at Coursera carried over intact into the other organizations he later founded. This shows both that he did not let go of the work of education itself even after leaving Coursera, and that the way he built courses together with people he trusted did not change much as he moved between organizations. The hands that built the Machine Learning Specialization together carried over into the hands that built the courses of the organization he founded afterward.

The foundation on which Coursera was able to reach learners across the world was not simply the single condition of internet access. Bound up together with it were the language

barrier to watching and understanding a lecture, whether one had spare money to receive a certificate, and whether the job market in a given country would recognize what was learned online. In the case of the young Syrian man discussed earlier, what he gained was not the lecture content alone. Because a certificate bearing Google's name came with it, the simple fact of coming from a country that had gone through civil war did not get him filtered out on paper, and he was able to land a position at a company in Germany. Knowledge crossed borders over the internet, but trust in the document proving that knowledge crossed borders only because it carried a name like Google or Stanford. This is why Coursera attached the names of universities and companies to each of its courses.

This value of a name was not something Coursera built by its own strength alone. The name Stanford carried trust built up over more than a hundred years beforehand, and the name Google carried a track record built up in Silicon Valley. Coursera took on the role of transferring this trust into the new vessel of the online classroom. It shared out the credit that universities and companies had built up over a long time to learners in difficult circumstances and to remote learners as well. What Andrew Ng achieved at Coursera was not creating new knowledge, but moving knowledge that already existed, along with the trust backing that knowledge, to those who had had difficulty obtaining it.

DeepLearning.AI: AI for Everyone

Even after Andrew Ng left Coursera, online education never disappeared from his work. If anything, it grew wider. In 2017 he founded a new education technology company called DeepLearning.AI. The mission this company set out is simple to summarize: to empower the global workforce to build an AI-powered future through world-class education, hands-on training, and a community where learners can connect with one another. The words sound grand, but what the company actually did was concrete. It built courses, lowered the cost so that as many people as possible could take them, and kept updating the content.

There is a history behind why DeepLearning.AI came to exist. In 2011, Andrew Ng released a machine learning course online from Stanford University, and more than 100,000 people registered for it. That moment confirmed that a course taken outside a university classroom could draw this many people. The following year, in 2012, he founded Coursera together with Daphne Koller, scaling this experiment up into a company. DeepLearning.AI was the next step. Where Coursera is closer to a marketplace gathering courses from many universities and institutions, DeepLearning.AI is a place centered on courses made directly by Andrew Ng himself and the team he assembled. From short, hands-on courses lasting just an hour to professional certificate programs spanning several months, more than 150 programs now carry this company's name. More than 7 million learners have passed through these courses. This figure is one DeepLearning.AI has disclosed, and it should be noted that it may vary depending on the point in time and the method of counting.

This education company did not target developers alone from the start. In fact, it put forward a course aimed at quite the opposite audience as its flagship work. AI for Everyone. This course is not aimed at engineers. It is aimed at people who have never written a single line of code — non-technical employees who need to collaborate with AI teams at their companies, managers planning products, marketing staff, and even executives who set a company's direction — people who need to understand AI but do not need to build it themselves. The entire course runs about six hours and fifty minutes, made up of thirty-five video lectures and four graded assignments. The reason this course exists is that it was built so that people could take it without knowing math or coding.

The course is divided into four weeks. The first week begins by asking what AI actually is. It sorts out what terms like machine learning, deep learning, data science, and neural networks actually mean, and, more importantly, draws the line between what today's AI can actually do and what it cannot. Much of the misunderstanding people have about AI starts from failing to distinguish between these two things. The goal of the first week is to close the gap between the imagined idea of AI thinking like a human and the actual technology of finding patterns in vast amounts of data.

The second week covers how AI projects actually run in practice. It shows the flow from an idea, through gathering data and training a model, to deploying it into an actual service. This part matters especially for non-technical learners. If the person who commissions or approves an AI project at a company has no idea how this process works, it becomes difficult to gauge how long it will take or what it requires. The third week covers how to establish AI within an organization — how to collaborate with an AI team, how to find opportunities within the company to apply AI to internal problems, and how to build a sustainable AI strategy that does not end with a single project. The fourth week turns its gaze beyond technology to society. It covers the discrimination and bias that AI can create, adversarial attacks designed to fool systems, the position AI occupies in developing countries, and the impact AI has on jobs. The fact that a course teaching technology hands over its entire final week to ethics and social issues says a great deal about this course's character.

Responses from people who took this course have also been recorded. One learner working as a software quality manager assessed that the course covered AI's core topics broadly while also providing a clear guide from choosing a project all the way through executing it. Another learner said that real-world examples and case studies helped them understand what practical impact AI has across various industries. What these assessments point to in common is one thing: complex technology was not explained with complex language.

If AI for Everyone was the course that lowered the threshold, the Deep Learning Specialization is for those who have crossed it. Called the Deep Learning Specialization, this program is an intermediate-level series made up of five courses, aimed at people who want to actually develop technology at the frontier of the AI industry. This program serves as a foundational course to help learners understand what deep learning can do, what its limits are, and how its results are used in actual industry. It runs on Python and TensorFlow and covers real applications such as reading medical images, autonomous driving, sign language recognition, music generation, and natural language processing. Rather than stopping at classroom theory, it keeps showing where this technology is actually being used.

In this course, learners build and train several core neural network architectures themselves. These include convolutional neural networks, which excel at processing images; recurrent neural networks, which handle sequential data; long short-term memory, which retains information without forgetting it even across long sequences; and transformers, which have become the foundation of today's large language models. On top of this, they also learn practical techniques that boost model performance — dropout, which prevents overfitting; batch normalization, which stabilizes training; and initialization techniques that set initial weights appropriately. These techniques are not concepts that appear only in papers but tools that solve problems encountered every time a model is actually trained. The scope covered by the Deep Learning Specialization reads as an attempt to narrow the gap between theory and

practice as much as possible.

As time passed, the center of gravity of AI technology shifted too — from an era of recognizing and classifying images to an era of generative AI that writes and produces answers as if conversing with a person. DeepLearning.AI released a new course to match this shift: Generative AI for Everyone. This course, too, is a beginner-level course requiring no coding skill, aimed at business leaders, professionals, and the general public. Andrew Ng himself teaches its roughly five hours of content, addressing how generative AI works, how far it can go, and where its limits begin.

This course is also divided into three weeks. The first week lays the foundations of generative AI and shows everyday uses such as writing, reading, and conversation. The second week goes beyond simply asking questions. It teaches techniques such as combining search results with the model to improve the accuracy of answers, and fine-tuning the model to suit a specific purpose, while examining the life cycle an AI project goes through from start to finish. The third week widens the view to cover generative AI's impact on business and society — questions like how to build responsible AI, and how the distant goal of artificial general intelligence relates to where we stand right now. A learner who completes this course becomes able to build a strategy for bringing generative AI into their daily work to raise productivity.

Another course released around the same time targets developers head-on: ChatGPT Prompt Engineering for Developers. This course was made in partnership with OpenAI and is a short, beginner-level course that delivers just the essentials in a mere one hour and forty minutes. Its goal is to help developers build faster, novel applications using large language models. The core skills the course covers are summarizing, which condenses text; inferring, which reads meaning from context; transforming, which converts one form of writing into another; and expanding, which enriches a short piece of writing. These four are basic tasks that anyone working with prompts encounters, and this course teaches how to craft sentences that carry out those tasks effectively.

DeepLearning.AI's course lineup did not stop here. Since AI technology changes on a scale of months, the company has kept producing short courses together with partner companies so that practitioners can quickly pick up new tools. The circumstances behind how each partnership came about, or the internal decision-making process at the company, cannot be confirmed from available records, but the resulting course list clearly shows the direction this company is heading.

First, there are courses that lower the barrier to entry into coding itself. AI Python for Beginners runs about eleven and a half hours and is designed so that even someone with no programming experience at all can learn to write, test, and debug Python code with the help of AI. By the end of this course, learners reach a level where they can build simple AI applications

on their own. The idea of using AI as an assistant in the very process of learning to code is an attempt to lower the threshold of coding education once again. A course in a similar vein is Spec-Driven Development with Coding Agents. Made in partnership with JetBrains, this course covers how to write clear specifications so that coding agents can build software that is maintainable. It shows that the ability to give precise instructions to an AI agent, rather than writing code directly by hand, is establishing itself as a new skill.

Next, there are courses that go deeper into the latest language models. Reasoning with o1 was made in partnership with OpenAI and teaches how to prompt and use OpenAI's o1 model for tasks that require complex reasoning. It covers how to plan, how to write code, how to reason about images, and even meta-prompting — writing prompts about how to design prompts themselves. Fast and Efficient LLM Inference with vLLM is an intermediate-level course made with Red Hat that covers optimizing open-source language models and using a tool called vLLM to actually deploy them and measure their performance. The scope of education now extends beyond building models to running those models fast and economically in an actual service.

The most recent trend consists of courses covering agentic AI — AI systems that judge and act on their own. Multi AI Agent Systems with crewAI, running about three hours, teaches how to design and direct a team made up of multiple AI agents using natural language. It is a method of automating a complex workflow that a single language model would struggle to handle alone, by having multiple agents with divided roles collaborate. Building Multimodal Data Pipelines was made in partnership with Snowflake and covers converting data such as images, audio, and video into text form that a language model can process, and building applications on top of that which handle multiple forms of data together. Agent Memory: Building Memory-Aware Agents, a course made with Oracle, teaches how to give agents a memory system that stores knowledge across multiple conversation sessions, retrieves it when needed, and refines it over time. Building agents that learn and improve on their own as time passes has now entered the realm of practical training. Build and Train an LLM with JAX was made in partnership with Google and is a course in which learners train a language model with twenty million parameters entirely from scratch using JAX, Google's open-source library. It differs from the other courses in that it offers the experience of building a model itself, even at a small scale, rather than stopping at learning how to use a massive commercial model.

Laid out this way, DeepLearning.AI's course lineup looks like a single staircase. A person who knows nothing about AI takes their first step in AI for Everyone; if they want to learn coding, they move on to AI Python for Beginners; if they want to build a solid foundation in deep learning, they take the Deep Learning Specialization; if they want to master the tools of the generative AI era, they go through the prompt engineering courses; and finally they reach the level of designing agent systems. Each step of this staircase is made up of a different partner

company, a different technology, and a different level of difficulty, but they all converge on one goal: to make AI technology not something only a handful of researchers can handle, but a tool anyone willing to invest interest and time can pick up.

Andrew Ng has compared AI to electricity on many occasions. A hundred years ago, when electricity was invented, that technology was at first a special force flowing only into a handful of factories and laboratories. But before long, electricity became infrastructure that flowed as a matter of course into every home and every factory. The courses DeepLearning.AI has produced can be seen as an attempt to realize this analogy in the realm of education. As long as AI remains a technology only a handful of experts can handle, that power cannot spread throughout society. Any single course may not be a great event in itself, but this entire lineup — spanning beginner to advanced, non-developers to developers racing to catch up with the latest technology — reads as one system trying to channel the power of AI into every corner of society.

Of course, it is too early to say this educational system is complete. Once a course comes out, a new model and a new tool appear a few months later, and DeepLearning.AI must chase after it again to build a new course. This is also why the list of recent courses for practitioners keeps growing. Keeping pace with the speed of technology is not a single achievement but a task that must begin again each time. Even so, the principle this company has kept to date has stayed consistent: not to pile more difficult language on top of a difficult concept when explaining it, and to keep a door open where anyone can start from their own position — someone who doesn't know how to code, someone who has just begun learning to code, and someone who already handles models in their day-to-day work.

The way this company builds its courses is itself worth examining, given how dense a course lineup DeepLearning.AI has assembled. Whenever Andrew Ng prepares a new course, he first settles on a partner company. Companies such as OpenAI, Google, Snowflake, Oracle, Red Hat, and JetBrains have all appeared on that list. Exactly what procedure these partnerships go through, or what negotiations took place before signing a contract with each company, cannot be confirmed from available records. Still, looking at the resulting courses, one pattern stands out clearly: the company that built a new tool joins hands with the educational body that teaches that tool. The course that optimizes vLLM was made together with Red Hat, which works with vLLM, and the course that trains a language model with JAX was made together with Google, which built JAX. When the place that builds a tool and the place that explains it are different, room opens up for information to lag or be distorted in between — and this company chose a method that closes that gap.

Thanks to this collaborative structure, even though any single course may have a short lifespan, the technology each course covers stays close to the latest version actually used in the field. That means learners can carry the tools they learn in a course directly over into their

own work. This is where it differs from the regular curriculum run in a university classroom. A university's curriculum is set on a semester or academic-year basis and does not change easily, but DeepLearning.AI's short courses appear and disappear at intervals of a few weeks to a few months. Some courses have the model they cover become an outdated version not long after release. Even so, this company has held to the approach of adding new courses on top rather than removing old ones — a structure that leaves options open both for learners who want to take an older course and for those who want to take the newest one.

The length of the courses is also a notable feature. There are courses running about six hours, like AI for Everyone, courses of just three hours, like Multi AI Agent Systems with crewAI, and courses of a mere one hour and forty minutes, like ChatGPT Prompt Engineering for Developers. The courses have been broken into portions a learner can finish within a day or two. For someone who has to study in stolen moments while working a job, a short course like this is a form they can actually complete, more so than a program spanning several months. This design aimed at raising completion rates is a different approach from the early online courses that simply carried university lectures over as they were. In exchange for breaking lectures into short pieces, the burden of having to release courses that much more often and in that much greater number fell squarely on the organization that is DeepLearning.AI.

Placing the flow of technology these courses cover in order also reveals which direction the AI industry has moved over the past several years. In the era when deep learning technology for classifying images and recognizing speech was one central axis, courses teaching convolutional neural networks and recurrent neural networks were at the center. Next, as large language models that generate sentences emerged, courses teaching how to write prompts increased. And most recently, courses have increased that cover a method in which multiple agents divide roles and collaborate rather than a single language model working alone, and a method in which an agent remembers past conversations and uses that memory to improve its judgment. The shift in the course lineup itself has become a record showing the chronology of AI technology. Whether Andrew Ng foresaw this flow in advance and planned the order of the courses, or made the courses by following tools that appeared first in the market, cannot be confirmed. What the resulting list clearly shows, however, is that education kept pace without falling behind each time the technology changed.

There is one more thing the courses DeepLearning.AI has made have in common. No course ends with a theoretical explanation alone — it always closes with a hands-on exercise or coding assignment. Even a course like AI for Everyone, which requires no coding, is set up with four graded assignments so that learners organize the concepts they've learned for themselves. The Deep Learning Specialization is always followed, after each theoretical explanation, by an assignment to actually implement a neural network. This structure appears to be a device meant to close the gap between listening to a lecture and actually being able to

use what was taught. Underlying the entire course design is the premise that a person who knows a concept and a person who can produce a result with that concept are different people.

The course lineup and hands-on structure alone are not enough to explain everything DeepLearning.AI does. Besides classroom-style education, this company has also published a weekly newsletter called The Batch. This newsletter is different in character from a course with a fixed curriculum and graded assignments. It centers on gathering that week's news from the AI industry and explaining newly released papers or models. Where a course, once made, keeps the same content for months or years, this newsletter deals with what happened that week or that month. This newsletter fills in the lag that a course cannot cover. From a learner's standpoint, a structure emerges in which they learn the fundamentals and skills through the regular courses and, through the newsletter, keep up with how far that technology has come at this very moment. The two move at different speeds but ultimately aim at the same purpose: keeping someone who has learned from stopping their learning.

The character of the writing this newsletter covers is also worth noting. Rather than simply listing announced facts, it comes with commentary pointing out what that news means in practice. There is a difference between an article that, when a new model is announced, conveys only that model's performance figures, and a piece that explains how that model will change the way developers actually work. The Batch leans toward the latter. It reads papers and press materials and then organizes and delivers what someone trying to bring that content into their own work needs to know. This connects to the attitude DeepLearning.AI has kept when making its courses — an attitude of first considering what the person receiving the information needs, rather than simply passing the information along as is.

There was one more thing beyond education and hands-on training in the mission DeepLearning.AI first put forward: building a community where learners can connect with one another. This part does not show clearly if you only look at the course descriptions, but it was there from the start in the mission statement the company itself put forward. An online course tends by nature to be something taken alone and finished alone. In many cases, it ends the moment you watch a video and submit an assignment. DeepLearning.AI tried to add to this structure a place where people meet people — a place where those taking a course exchange questions with one another, work through sticking points together, and go on to form relationships that help each other's careers. Without such a place, a course remains merely a channel for delivering knowledge. With such a place, a course also functions as a channel connecting person to person. The fact that the company placed this community ahead of the courses themselves in its mission statement reads as meaning that this organization considered relationships between people to be a goal as important as delivering knowledge.

What remains for a learner after finishing a course is also worth examining. The Deep Learning Specialization and various professional certificate programs come with a certificate of completion once finished. This certificate is not a credential guaranteed by a state or a university the way a degree is. Still, it is used as evidence on a résumé or in a job search to show what skill a person has learned to what level. If a person in charge of hiring finds a DeepLearning.AI program name on an applicant's résumé and can guess what that name means, then this certificate is doing its job. If, on the other hand, no one recognizes that name, the certificate amounts to no more than a piece of paper. One reason DeepLearning.AI has carefully designed the content and difficulty of each individual course is, in the end, that this certificate must become a mark that earns trust in the field. Building a course and protecting the value of that course's name are not separate matters.

Bringing all of this activity together makes the position of an organization called DeepLearning.AI somewhat clearer. This company is less a place that sells courses than a place that, through several devices — courses, a newsletter, a community, and certificates — keeps people staying close to the technology of AI. A person who takes one course and stops, and a person who takes a course, subscribes to the newsletter, and keeps talking with other learners within the community, end up standing in different places after the same amount of time has passed. It is hard to confirm from remaining records in what order, or what Andrew Ng decided first, when designing these various devices. But looking at the results that remain now, there are traces of adding, one at a time, a different kind of device for every place that a single format — the course — could not fully fill. Keeping a learner from stopping their learning: what DeepLearning.AI has repeatedly attempted through its various outputs narrows down, in the end, to this one thing.

Laying out the course list also brings another marking into view: the difficulty labels of Beginner and Intermediate. AI for Everyone is marked Beginner, and so is Generative AI for Everyone. ChatGPT Prompt Engineering for Developers is classified as a short beginner course. Deep Learning Specialization, on the other hand, is marked Intermediate, and so is Fast and Efficient LLM Inference with vLLM. Going by this labeling alone, the structure is one where courses that require no coding or math sit on the Beginner side, and courses that involve implementing a neural network directly or optimizing and deploying an open-source model sit on the Intermediate side. What stands out is that no course anywhere on this list carries an Advanced label. Even a course on designing multiple agents, or a course on training a language model from scratch, is listed without any official difficulty label. Why the company chose this labeling approach, and what discussions took place internally, cannot be confirmed from available records. Going only by the result, though, it reads as a structure that, rather than stacking difficulty in layers to filter learners out, opens the door with only two tiers — Beginner and Intermediate — and leaves it open above that for anyone interested to choose their own way in. Even though the technology involved in designing agents or training a

language model is itself complex, no separate formal gate has been set up that one must pass through to take that course.

What standard this difficulty labeling actually follows shows up a bit more concretely within the Deep Learning Specialization. The initialization techniques this program teaches have names: Xavier initialization and He initialization. Depending on what numbers are put into each layer's weights when a neural network is first built, training can either proceed smoothly or get stuck from the very start. If the weights are filled with values that are too small, the signal fades each time it passes through a layer, leaving almost no information by the later layers; conversely, if filled with values that are too large, the signal balloons each time it passes through a layer, making the computation unstable. Xavier initialization and He initialization are methods that, between these two extremes, calculate an appropriate range matched to the size of the layer. Dropout and batch normalization occupy a similar position. Dropout randomly turns off some of the neural network's neurons during training so that the model does not depend on any single path, and batch normalization realigns the distribution of values entering each layer during training to stabilize the pace of learning. The fact that these names do not remain terms confined to papers but become something learners run into directly while writing code within the course helps explain why the Intermediate label was attached.

The intermediate-level course covering vLLM carries a similar character. What this course teaches is taking an already-built open-source language model, optimizing it to respond faster, actually deploying the result, and measuring processing speed and resource usage. Building a new model and running an already-existing model at a speed a service can bear require different abilities. The latter is not an area easily accessible without knowing the process of training a model, because one needs to understand what computation the model goes through each time a single request comes in, in order to judge where time can be cut. That is why this course is aimed less at someone writing Python code for the first time and more at a developer who has already worked with models. Meanwhile, even within the same list of recent courses for practitioners, a course designing multiple agents with crewAI, or the multimodal pipeline course made with Snowflake, carries no separate official difficulty label. Even though the technology they cover is new and unfamiliar, these appear to be courses centered on giving instructions in natural language, so no separate barrier to entry was set.

Seen this way, the two tiers DeepLearning.AI has labeled — Beginner and Intermediate — follow a standard different from how recent the technology is. A course released recently is not necessarily a difficult course, and the Deep Learning Specialization, which has been around since early on, in fact remains at the higher tier. The standard that divides difficulty lies not in when the technology a course covers came into the world, but in what knowledge a learner needs to already have in order to learn that technology. A course where you must code a

neural network from scratch and a course where you give instructions in natural language to an already-built tool require different preparation even if they came out in the same period. Even as DeepLearning.AI's course list grew over several years, the Beginner tier was never left empty. AI for Everyone took on that role, and the generative AI course and the introductory Python courses that came after it were placed side by side at the same tier. It means that even as technology changed, a place was always kept ready for someone learning for the first time.

This two-tier labeling system serves as a small compass for someone trying to choose a course. It reduces the chance that a person who has never coded before opens a course marked Intermediate first and gets flustered, and it also reduces the chance that someone who has already trained a model wastes time on a beginner course. Behind the fact that reviewers said each course had a clear roadmap, this labeling was likely playing a part. However well a lecture is put together, if a learner cannot find the entrance suited to them, that course never even gets started. This is likely also why DeepLearning.AI, even while continuously making new courses, has kept this simple two-tier system of Beginner and Intermediate steady without disturbing it.

The pace was different. The courses changed too. The circumstances of the people learning changed as well. Ng did not bundle that difference into one long school. He divided it into short doors instead.

Landing AI and Artificial Intelligence on the Enterprise Floor

Andrew Ng, who had taught the world in the language of education, started a company again. This time his student was not an individual learner but the enterprise itself. His task was to turn concepts learned in a lecture hall into forms that factories, offices, hospitals, and law firms could actually use. The company he founded was named Landing AI, and he took his place as its chairman and chief executive officer.

After founding Landing AI, the first thing Ng picked up was not a new technology but a single document. He personally wrote a guide called the 'AI Transformation Playbook.' At the opening of this document was a sentence he had long been sharpening. When the internet first entered the world, countless companies built a single website and called themselves internet companies. Ng aimed this exact misconception squarely. Putting a website on a store does not make it an internet company. He carried the same logic straight over to AI. Bolting one piece of deep learning technology onto an ordinary traditional company does not make that company an AI company. These two sentences are short, but they show clearly what he was guarding against on the enterprise floor. Adopting technology that only changes appearances and a transformation in which the whole organization actually works differently are not the same thing.

The scope the playbook covered was not limited to a single technology. It addressed securing early wins through pilot projects, how to build an organization capable of handling AI, what strategy to use in gathering and organizing data, and how to build trust inside and outside the organization throughout this entire process. He did not leave AI transformation as the responsibility of a single technical department alone. He believed that the judgment of management, the cooperation of people on the ground, and the discipline of handling data all had to change together as the company's overall constitution shifted. Buying new technology and changing the way work actually gets done with that technology were tasks of different magnitude, and he never conflated the two.

The place that embodies this philosophy is today's Landing AI. The company's mission, condensed into one sentence, reads: make the world's documents computable. Behind this seemingly modest goal lies a question Ng had held onto for a long time. In an age overflowing with cutting-edge technology, what actually holds most companies back is not a dazzling algorithm but tedious, mountainous piles of documents. Contracts and invoices, insurance policies and medical records, shipping slips and legal documents still often have to be processed one page at a time by a human being. Ng made this old bottleneck the company's central challenge.

Today's Landing AI is not a place Ng leads alone. He holds his position as founder and chairman of the board, while day-to-day management is handled by CEO Dan Maloney. Investors such as Intel Capital and the AI fund McRock Capital back the company. This means that a name that began in academia has now become an axis supporting a business organization that investors and management move together.

The documents Landing AI faces are not clean. The documents real companies handle run across multiple pages, formats differ from company to company, and exceptions that break the standard appear endlessly. This is messy data, different in texture from the neatly curated sample data of a research lab. Reviewing such documents by hand had long been taken for granted at financial institutions and insurance companies. Landing AI sought to replace this manual review with agentic AI. The word agentic means that instead of spitting out a single answer and stopping, the AI reads a document, makes judgments, and carries a series of tasks forward to the next step on its own.

The technology that realizes this is Agentic Document Extraction. Rather than one large monolithic box, this technology is divided into several finely segmented functions. Parsing, which reads in a document; extraction, which pulls out the needed information; schema building, which sets up a framework matching the document's structure; classification, which identifies the document type; sectioning, which divides a document by category; and splitting, which breaks a long document into manageable units — each step is provided as its own separate API. A company can pick and attach only the pieces that fit its own work. The technology built this way has reached out to financial services and insurance, healthcare, energy and utilities, legal, and logistics — anywhere documents pile up.

The point where Landing AI sets itself apart from other AI startups is clear: the phrase production, not prototype. A demo that looks plausible only in a promotional video and a system embedded in a real team's daily workflow that produces measurable results are entirely different things. Ng aimed at the latter. The second principle he emphasized was trust by design. Automation is not completed by accuracy alone. In high-stakes industries — that is, industries where a single wrong judgment can lead to large losses or legal liability — the results AI produces must be verifiable, evidence-based, and auditable. Landing AI did not stop at pulling values out of documents; it tried to go further, turning those values into reliable inputs that actually drive downstream systems and decisions. The phrase 'from documents to systems' is a condensed expression of this direction.

Evidence that this direction did not remain mere sloganeering comes from the scorecard. On DocVQA, a benchmark that measures the ability to answer questions by looking at a document, Landing AI's technology scored close to 99.26 using text alone, without images. P.K. Mishra, founder of Proxima, evaluated that in a benchmark comparing several document extraction tools, Landing AI's Agentic Document Extraction was the only tool that scored high

across all four evaluation criteria. The same assessment held outside the lab as well. The head of data and analytics at a Fortune 100 financial services company said the system delivers perfect traceability. He added that this traceability was precisely the requirement that made it possible to actually deploy this technology in a tightly regulated real-world environment.

This question of trust revealed itself even more clearly the more complex the system became. When building a plan review agent that had to reason across a knowledge graph of building codes, crossing multiple drawings and multiple specialties, the wall developers ran into kept turning out to be the same spot. If the information pulled from a page cannot be trusted, no matter how sophisticated the reasoning stacked on top of it, it is useless. However dazzling the algorithm, if the input wavers, the whole system wavers with it. This is why Landing AI poured so much effort into document extraction alone. Ng chose to hold onto the unglamorous foundation — the reliability of data — over the more visible algorithm.

Landing AI left this technology open so that anyone could try it out in a few minutes without registering a credit card. Anyone can directly watch the process by which a single complex document, once uploaded, is turned into structured data. The man who once taught concepts in a lecture hall now has an enterprise representative handle the output with their own hands. The way of teaching changed, but the attitude of lowering the threshold to unfamiliar technology carried straight through.

Ng, who once ran experiments scaling up neural networks at a giant tech company and spread lectures across the world beyond the walls of a university, is now face to face with stacks of paper that had lain dormant in corporate drawers. In choosing a tedious operational problem over a flashy research achievement, Landing AI traces the arc of his path — from academia to industry, and then to the least conspicuous corner of industry itself. This attitude continues at the other organizations he has since built, whether the AI Fund that supports startups or a seat on the board of a company like Amazon. Ng sought to build one system that actually works before making any grand, glamorous promise.

Looking a little more closely at the scope of automation Landing AI handles reveals that the company never aimed at a single industry from the start. Ng chose the subject of documents itself. Few businesses, in any sector, escape the accumulation of contracts, invoices, applications, and supporting records. Rather than becoming an expert in one particular industry, he chose to dig into a single bottleneck common across industries. This choice echoes both his time at Google Brain immersed in experiments scaling up neural networks and his time at Coursera opening a single course simultaneously to students in many countries around the world. In either case, he worked not by digging deep into a narrow specialty but by finding the common wall that was blocking many places at once.

That Agentic Document Extraction is broken into several small functions is also of a piece with this way of thinking. The six pieces — parsing and extraction, schema building and

classification, sectioning and splitting — are each opened up as an independent API, so a company can pick only the parts it needs and attach them to its system. An insurance company can use only the function that classifies claims, and a law firm can use only the function that divides a long contract by clause. This structure differs from a trap that technology vendors commonly fall into — forcing a customer to buy one giant box whole. Ng took as given that every company's situation differs, and designed the tool to break into small pieces that could absorb that difference.

Behind the fact that this approach was even possible lies a principle Ng has long emphasized: the idea that good data, more than a good algorithm, determines a system's actual performance. Academic benchmarks generally compete for scores on neatly organized samples. But the documents in a real company's drawers are unevenly scanned, mixed with handwriting, and the same item is called by a different name at every company. This is why Landing AI kept emphasizing, rather than just laboratory benchmark scores, its ability to handle the messy data that comes out of real operating environments. A single number like accuracy cannot explain this difference. Whether a system can keep producing results without breaking down when it meets an unfamiliar format or an exceptional case is a far more important yardstick on the ground.

The principle of trust by design takes on meaning again at this point. Ng did not treat automation purely as a matter of accuracy. He believed that a human had to be able to go back and check the path by which a result was reached. Verifiable means there is a way to check the result again; evidence-based means the result comes from the document's actual content and is not a guess the system fabricated. Auditable means that someone must be able to trace back, even later, why a given judgment was reached. In industries with dense regulation like finance and insurance, whether a given result can be explained as trustworthy mattered more than whether that one result was accurate. The three conditions Landing AI puts forward were an answer that met this industry's long-standing demand head-on.

That this principle did not remain an abstract slogan is confirmed again in the testimony of a financial services company among the Fortune 100. When that company's head of data and analytics mentioned perfect traceability, it meant that more than the accuracy of the result alone, being able to trace back the process by which that result was produced was the standard that decided actual deployment. The more tightly regulated the environment, the more adoption is decided not by a single answer the system produces but by whether the path leading to that answer is transparent. Landing AI concentrated the company's resources on satisfying this condition.

The case surrounding the plan review agent shows, from a somewhat different angle, why this principle was necessary. Building codes are a domain where regulations across multiple drawings and multiple specialties are tangled together. Plumbing drawings, electrical

drawings, and structural drawings each follow different rules yet must fit together within a single building. In building such a system, the wall developers ran into was not a lack of flashy reasoning algorithms. It was the simple fact that if a single piece of information pulled from a page is wrong, the entire result wobbles no matter how sophisticated the reasoning stacked on top of it. This is a lesson anyone who designs a system eventually runs into. Data quality at the less visible lower layer, more than the more visible algorithm at the upper layer, sets the ceiling for the whole system. The reason Ng poured the company's capability into the outwardly unglamorous task of document extraction appears to be that he confirmed this lesson repeatedly in practice.

The list of industries Landing AI targeted — financial services and insurance, healthcare, energy, legal, and logistics — looks distant from one another on the surface. But these industries share something in common. Each has accumulated paper documents and scanned files for a long time, and handling those documents still depends heavily on human hands and eyes. Contract review at law firms, claims examination at insurers, and customs paperwork processing at logistics companies are all tasks that require skilled staff to read documents one page at a time. This work is neither glamorous nor often in the media spotlight, but it is a spot that actually governs a company's labor costs and processing speed. Ng pinpointed exactly this inconspicuous spot.

This choice also connects to the attitude he showed at Coursera and DeepLearning.AI. He always searched for the point where the greatest number of people were stuck and opened a door there. In online education he lowered the physical threshold of the university lecture hall; at DeepLearning.AI he translated concepts that only specialist researchers understood into the language of practitioners. At Landing AI, that threshold was not the technology itself but the shape of the data. Turning information trapped in the form of a document into a structure a system can handle — that is the actual content behind the phrase Landing AI keeps repeating, making it computable.

The attitude Ng showed through this company ultimately converges on one point: the principle that adding new technology and making an organization actually work differently are not the same thing. It is no coincidence that Landing AI left open a tool letting anyone test a single document without registering a credit card. The experience of a company representative putting in their own document and checking the result builds conviction far faster than trust explained in words. Ng has known since his Coursera days that the shortest path to building this conviction is to let people try it themselves, and he carried the same approach straight over to Landing AI.

The industries Landing AI handles each carry a different intensity of regulation, and that regulation even determines how they treat documents. Financial regulators demand that the grounds for a review be traceable even years later, and insurance regulators demand

disclosure of where in the document the reason for approving or denying a claim came from. The contracts law firms handle are documents where a difference in interpreting a single clause can escalate into litigation, and the records healthcare institutions keep are directly tied to patient safety. In these industries, a system's ability to produce an answer quickly is a secondary matter. The ability to explain where that answer came from comes first. Behind Landing AI putting forward the three conditions of verifiability, evidence basis, and auditability side by side lies this overlap of industry-specific demands.

The reason behind the name agentic is worth revisiting here as well. Conventional document processing automation commonly recognized only documents that matched a fixed template and handed off anything outside that template to a human. The agentic approach Landing AI emphasizes breaks the process of reading a document itself into several stages, making a judgment at each stage before moving to the next. In the parsing stage it reads the document's physical structure; in the classification stage it identifies what kind of document this is; in the schema-building stage it sets up a framework fitting that document type; and in the extraction stage it pulls out values matching that framework. This flow resembles the procedure a person follows when reviewing a document — first skimming the document type, deciding which items to look at, and then finding and transcribing the needed values in that order. Landing AI did not package this procedure into one single box but kept it open stage by stage, so a company could plug the system in at the point that matched its own review process.

The reason this approach was needed is that document processing work originally required human judgment. Whether to approve a single invoice is not a problem solved by simple character recognition alone. Whether the billed amount matches the contract terms, whether the attached documentation meets regulations, and whether it looks unusual compared to past history all have to be weighed together. The reason Landing AI set up classification, sectioning, and splitting as separate functions after extraction was to bring this kind of supplementary judgment inside the system. It is work that does not stop at pulling a value out of a single document but also organizes the context around that value so that the next system down the line can pick it up and use it.

The score in the high 99-point range on DocVQA shows that this technology reads text accurately, but that number alone does not decide whether a company adopts it. The phrase perfect traceability mentioned by the representative of the Fortune 100 company points to the fact that accuracy and trust are yardsticks on different levels. Accuracy refers to the rate at which a system produces the correct answer; traceability refers to whether one can trace back where that answer came from. In regulated industries, without the latter, no matter how high the former is, it is hard to put the system into actual operation, because when an auditor or regulator later asks how a given judgment was reached, the system must be able to show

which sentence in the document the value was drawn from. Landing AI's phrase 'from documents to systems' reads as meaning that it hands over each extracted value along with its source attached.

The wall the developers of the plan review agent ran into is another face of the same story. Architectural drawings are divided across multiple sheets, and plumbing, electrical, and structural elements each follow different regulations yet must interlock within a single building. In such a system, no matter how sophisticated the reasoning stage is, if there is even one number misread from a drawing at the outset, every judgment built on top of it leads to a wrong conclusion. What the developers confirmed was that, in the end, not the brilliance of reasoning but the accuracy of extraction sets the ceiling of the system. This is why Landing AI concentrated the company's resources on the outwardly unglamorous task of document extraction.

Looking again at the list of industries Landing AI targeted, the six items — finance and insurance, healthcare, energy, legal, and logistics — appear far apart from one another on the surface. Yet all of them share the common trait that paper documents and scanned files have piled up over a long time, and handling those documents still depends heavily on human hands and eyes. Contract review at law firms, claims examination at insurers, and customs paperwork processing at logistics companies are all tasks requiring skilled staff to read documents one page at a time. This work is inconspicuous but is the actual spot that governs a company's labor costs and processing speed. Ng pinpointed exactly this inconspicuous spot.

This choice also connects to the attitude he showed at Coursera and DeepLearning.AI. He always searched for the point where the greatest number of people were stuck and opened a door there. In online education he lowered the physical threshold of the university lecture hall; at DeepLearning.AI he translated concepts that only specialist researchers understood into the language of practitioners. At Landing AI, that threshold was not the technology itself but the shape of the data. Turning information trapped in the form of a document into a structure a system can handle — that is the actual content behind the phrase Landing AI keeps repeating, making it computable.

The attitude Ng showed through this company ultimately converges on one point: the principle that adding new technology and making an organization actually work differently are not the same thing. It is no coincidence that Landing AI left open a tool letting anyone test a single document without registering a credit card. The experience of a company representative putting in their own document and checking the result builds conviction far faster than trust explained in words. Ng has known since his Coursera days that the shortest path to building this conviction is to let people try it themselves, and he carried the same approach straight over to Landing AI.

While Landing AI stays fixed on a single subject — documents — one condition Ng attached to the company stands out. The phrase production, not prototype. The reason he insisted on putting this phrase up front is not unrelated to something that commonly happens in the AI startup world. No small number of systems run smoothly at a demo but grind to a halt once they enter a real company's day-to-day work. A single staff member demonstrating in front of a screen gets clean answers, but the system collapses the moment hundreds of people push in documents of different formats every day. Ng set this gap as the company's first criterion for judgment. It means that only a system delivering measurable results across an entire team and workflow is recognized as a product.

This standard also connects to a habit from his time in academia. A research paper only needs to show that a new method scored better than existing ones. But a system deployed inside a company does not end with a single score. It has to withstand this morning's newly arrived document in an unfamiliar format, a poorly scanned document, and even a document with a staffer's handwritten notes added in. The reason Ng, at Landing AI, deliberately set real-company testimony alongside benchmark scores is that the two kinds of verification tell different things. A benchmark shows how accurate a system is; an actual deployment case shows whether that accuracy holds up even in an unfamiliar environment.

Looking at the breadth of industries Landing AI covers also reveals that this is not a company selling a single piece of technology under several different names. Finance and insurance handle claims and approvals; healthcare handles medical records and prescriptions; energy and utilities handle equipment inspection reports. Legal handles contracts, and logistics handles customs paperwork. These six fields differ in the shape of their documents and in the intensity of their regulation. And yet the reason Landing AI could cover all of them with a single technology is that Ng aimed from the start not at the language of any specific industry but at the common form of the document itself. This decision seems to have started not from a strategy he devised after surveying multiple industries in advance, but from the simple observation that documents themselves pile up regardless of industry.

The fact that Agentic Document Extraction is provided as six separate APIs also fits with this observation. The pieces — parsing and extraction, schema building and classification, sectioning and splitting — can each be called independently. If an insurer needs only the function that classifies claims, it can take just that function. If a law firm needs only the function that divides a long contract by clause, it can attach just that piece. This structure is a design that presupposes each company already has a different system in place. Rather than demanding that a company tear apart its entire system to fit Landing AI's way of doing things, Landing AI instead fits itself into the company's existing way of working.

Behind this flexible structure sits an idea Ng has held onto for a long time: the idea that good data, more than a single good algorithm, determines a system's actual performance.

Academic benchmarks generally compete for scores on curated samples, but the documents in a company's drawers are unevenly scanned, mixed with handwriting, and the same item is called by a different name at every company. This is why Landing AI kept emphasizing, rather than just laboratory scores, its ability to handle messy real-world data. A single number like accuracy cannot explain this difference. Whether the system can keep producing an answer without collapsing when it meets an unfamiliar format or an exception is a far heavier yardstick on the ground.

The principle of trust by design takes on meaning again here. Ng did not treat automation purely as a matter of accuracy. He believed a human had to be able to go back and check the path by which a result was reached. Verifiable means there is a way to check the result again; evidence-based means the result comes from the document's actual content and is not a guess the system fabricated. Auditable means that someone must be able to trace back, even later, why a given judgment was reached. In industries with dense regulation like finance and insurance, whether a given result could be explained as trustworthy mattered more than whether that one result was accurate.

The testimony of a financial services company among the Fortune 100 shows that this principle did not remain mere sloganeering. When that company's head of data and analytics mentioned perfect traceability, it meant that more than the accuracy of the result alone, being able to trace back the process by which that result was produced was the standard that decided actual deployment. The more tightly regulated the environment, the more adoption is decided not by a single answer the system produces but by whether the path leading to that answer is transparent.

The case surrounding the plan review agent shows, from another angle, why this principle was necessary. Building codes are a domain where regulations across multiple drawings and multiple specialties are tangled together. Plumbing drawings, electrical drawings, and structural drawings each follow different rules yet must fit together within a single building. In building such a system, the wall developers ran into was not a lack of sophisticated reasoning algorithms. It was the simple fact that if a single piece of information pulled from a page is wrong, the entire result wobbles no matter how sophisticated the reasoning stacked on top of it. Data quality at the less visible lower layer, more than the more visible algorithm at the upper layer, sets the ceiling for the whole system. The reason Ng poured the company's capability into the outwardly unglamorous task of document extraction appears to be that he confirmed this lesson repeatedly in practice.

The decision to leave open a tool letting anyone test a single document without registering a credit card is also not unrelated to this principle. The experience of a company representative putting in their own document and checking the result builds conviction faster than trust explained in words. Back in his Coursera days, when he opened online courses to students

around the world, Ng showed the course material up front and let students decide for themselves whether to enroll. The same sequence repeats at Landing AI. Show the result first, and let trust be built by the person who has seen that result. The setting moved from a lecture hall to a company, but the way of reaching out first to a wary stranger has not changed much.

AI Fund, Amazon, and the Startup Ecosystem

What Ng confirmed at Landing AI was the sheer weight of transforming a single company. Even turning one manufacturing plant's document-processing department over to AI required a long stretch of time and painstaking work. That experience naturally led to the next question: more founders would keep emerging who wanted to change the world with AI, so what would they lack? Ng's answer was not simply capital. It was an eye for validating ideas, an instinct for assembling an early team, and the practical knowledge needed to turn technology into an actual business. AI Fund was the organizational form that answer took.

AI Fund chose a different path from the ordinary venture capital model of putting money into startups in exchange for equity. It calls itself a venture studio. Where venture capital picks companies that already exist and invests in them, a venture studio builds a company together with the founder starting from the idea stage. Ng led this organization as managing general partner, holding to a method of walking alongside founders through the process of finding an idea, validating the market, and building a sustainable business. AI Fund's focus was not on companies that had already found their footing but on the earliest stage of all, before even a pitch deck existed. The judgment was that this was the stage where help could make the biggest difference.

The names behind the capital that backed this judgment are familiar ones in Silicon Valley's venture industry. New Enterprise Associates, Sequoia Capital, and SoftBank Group all participated as limited partners. AI Fund later raised a second fund, so-called Fund II, meant to co-found new AI startups; it totaled 190 million dollars and closed oversubscribed, beyond its target amount, meaning major institutional investors competed to put money in. Underlying this organization is a line Ng has repeated since his Google Brain days: AI is the new electricity. Just as electricity spread across industry and made countless new products and services possible, the belief is that AI will power tens of thousands of new applications. AI Fund was an attempt to realize that belief one startup at a time.

Why a venture studio rather than venture capital? Whatever personal deliberation or comparison with other investment models lay behind this choice is not found in public records. What is clear, however, is the structure and mode of operation of the organization that resulted.

The list of companies AI Fund has nurtured shows that its targets were not narrow. There is RapidFire AI, which builds experiment management tools, and Aligned AI (Olakai), which helps companies manage their use of AI tools safely. There is also Steam, which has set the goal of making literacy a basic right for every child. Octagon, which finds hidden opportunities and risks for investors, and Rippl, which helps middle managers who easily lose visibility lead their teams with confidence, are also on the list. Skyfire AI, an autonomous drone platform that

carries out rescue operations in disaster situations like hurricanes, is evidence that AI Fund did not stay confined to consumer services. The list crosses education, disaster response, and enterprise software, all the way to SpeechLab, which aims to lower language barriers with voice-to-voice translation technology; Sunrise AI, which builds customer intelligence for sales and marketing teams; Warpa, which wants to turn a job into a career passion through AI-driven career guidance; and Workhelix, which maps out an AI productivity path for large enterprises.

AI Fund's reach also extended beyond the United States. It announced an expansion into Taiwan through an Asia hub and carried out AI innovation work in partnership with Mitsubishi Corporation. It formed a partnership with the energy company AES to jointly build an AI energy startup within six months. What personal circumstance, if any, started this regional expansion and industry partnership for Ng is not documented. What can be confirmed is only the fact that AI Fund extended its reach beyond the U.S. consumer market into Asia and the energy industry.

Alongside growing startups, Ng also turned his hand to transforming large companies that were already established. Where Landing AI helped bring AI into manufacturing settings, AI Aspire is an advisory organization aimed at large enterprises more broadly. Ng founded the company together with colleague Kirsty Tan and serves as managing partner. The problem he identified in an interview was clear: to actually shift the landscape of AI adoption, it is not enough to draw in developers or individual consumers — large businesses have to be moved. Embedded in that judgment was the view that no matter how widely a single consumer chatbot spreads, it cannot shift an entire industry's productivity curve.

To pursue this goal, AI Aspire formed a partnership with a team from the global management consulting firm Bain & Company. The collaboration is led on Bain's side by Christoph, and the two organizations work directly with the executives of large corporations to provide advisory services. What they are aiming at lies beyond an organization's small-scale experiments with AI. Succeeding at a single pilot project is a different matter from scaling that success across the whole organization to actually change outcomes. The strategic goal AI Aspire has set is to produce exactly this kind of scalable, transformative result. Whatever deliberation Ng may have gone through in reaching this decision, or any internal disagreement about partnering with a major consulting firm, cannot be confirmed from public records.

On April 9, 2024, a new position was added to Ng's résumé. He joined the board of directors of Amazon, the world's largest e-commerce company and a major cloud computing company. The seat opened up because Judy McGrath, who had served on the board for ten years, chose not to stand for re-election at that year's annual shareholder meeting. What conversations passed between Amazon and Ng before the offer was extended and accepted, or what

personal connections were involved, are not known. What can be confirmed is only the outcome: that he became a member of the board.

Amazon's official statement laid out the reasoning behind this recruitment. The company described artificial intelligence, generative AI in particular, as one of the most transformative technologies of this era, and explained that Ng's experience was needed to help the board understand the opportunities and challenges this technology opens up. The statement listed his background in detail: a researcher who has written more than 200 papers in machine learning and robotics, the leader who founded Google Brain, someone who served as chief scientist at Baidu, and the founder of Coursera and Landing AI. The fact that both his academic research record and his business experience across multiple companies were cited together shows that Amazon expected more of him than a single narrow technical advisory role. The board stated that his insight would help it form a company-wide perspective on the changes AI will bring to society and business.

If there is one word that ties together the roles Ng has taken on across various organizations, it would be mentor. Through AI Fund, he has tried to offer support that goes beyond the simple relationship between investor and founder. The attitude he has described is this: he holds entrepreneurs who believe in what they are doing and work to make it real in high regard, and considers it a core value to help them build the right team under their own leadership. This suggests that Ng has positioned himself not simply as a source of capital but as someone who shares technical knowledge, organizational experience, and business judgment. Whatever conflicts or emotional exchanges he may have had with founders in the course of mentoring are not addressed in public records.

Where this mentorship becomes concrete is in the Insights channel and blog that AI Fund runs. There, practical guidance has piled up that early-stage founders can put to immediate use. Posts cover growth marketing methods for early-stage AI startups, how to build a pitch deck that draws funding from investors, how to hire a company's first salesperson, and the three pillars a startup uses to tell its own story. The channel also addresses the problem of building a technical organization: what founders need to know before hiring a technical team, a forecast of how organizations will change once every team member can code, and analysis of the tradeoffs that come up when assembling a large language model tech stack. There are also posts about the industry itself: insights drawn from AI transformation through venture studio and corporate partnerships, and a piece on where traditional vector search fails in enterprise environments and the alternative of knowledge graphs. These channel posts are published free of charge, so founders who have received no direct investment from Ng's organization can access them as well.

Ng did not confine this knowledge-sharing to writing alone. There is also a case where he invited Pandora founder Tim Westergren for a masterclass conversation on the theme of

perseverance and leadership. It was an attempt to pass on, directly to the next generation of founders, the experience of someone who had spent a long time building a company. AI Fund recruits applicants through a program called Engineer in Residence and holds Builder events in Mountain View to meet the next generation of AI founders face to face. The goal Ng has set out at these gatherings can be summed up in a single line: building great companies that advance humanity. What personal advice Ng gave participants at these in-person events is not recorded in the materials available.

Ng's activity as a mentor did not stay confined to corporate settings and the startup ecosystem — it reached into the realm of policy as well. He submitted a written statement to the AI Insight Forum organized by the U.S. Senate, offering his views on national AI policy and the direction of regulation. The experience he built up as a researcher, entrepreneur, and corporate board member became material for policy discussion. This wide range of activity also drew outside recognition, and in 2023 his name appeared on Time magazine's list of the 100 most influential people in AI.

The trajectory revealed in this chapter has a different texture from Ng's earlier path. Where Google Brain was about large-scale research, Coursera and DeepLearning.AI about education, and Landing AI about transforming individual companies, AI Fund, AI Aspire, and the Amazon board involved tools outside technology itself — capital, boards, and policy. The earlier pattern, of moving ideas validated in the lab into lectures and then moving the needs of people he met through those lectures into problems on the corporate front line, gained new channels here in capital and organization. Yet what results these channels actually produced — the survival rate of portfolio startups, or the concrete performance of the companies AI Aspire has advised — cannot be confirmed from public materials. What can be confirmed is only the structure of the organizations, the roster of participants, and the goals they have set out. Being clear about this limitation is necessary to understand Ng's activity in this period without exaggeration.

The real dividing line between venture capital and a venture studio is the timing of when the money comes in. Traditional venture capital appears after a founder has already started a company and even built an early product. The investor reviews the business plan and early metrics and, if satisfied, hands over funds in exchange for equity. Naming the company, finding a co-founder, meeting the first customer — the founder has often already finished all of this alone. The approach AI Fund took moves this sequence earlier. From the point when an idea has not even been written up on a single page, when all that exists is a hypothesis-level notion the founder jotted down alone in a notebook, the organization steps in and helps build the company's framework together. In this structure, the line between investor and founder blurs. AI Fund's own people join in early market research, help write the first job postings, and work alongside the founder to define the problems the first version of the product needs to

solve. That Ng leads this organization under the title of managing general partner can be read in this light. The role is less that of an investment committee member and more that of a co-founder simultaneously involved in the founding process of several companies.

The single fact that Fund II closed oversubscribed, beyond its target amount, is worth examining on its own. In the venture industry, oversubscription means more money flows in from investors than the fund intended to raise, forcing the manager to return some of it. It means the fund ended up in the position of choosing among capital rather than being a startup short on funds. This kind of result is not common. The venture studio model itself is newer compared to traditional funds, and there is relatively little data to verify its track record. The fact that limited partners — institutional investors who put money into a fund but leave the operating to someone else, such as New Enterprise Associates, Sequoia Capital, and SoftBank Group — put capital into this model reads as a signal within the industry. These three institutions have each, in their own way, held a long-standing place in Silicon Valley. That such institutions moved to validate it can be read as meaning that it was not simply Ng's personal name recognition but the operating method of the organization he built that was recognized as a repeatable structure.

Looking again at the list of portfolio companies reveals a pattern. Literacy education, disaster-rescue drones, voice translation, sales support software, coaching tools for middle managers, investment risk analysis — these companies differ in industry, in customer, and in how the technology is used. This diversity is not accidental; it connects directly to the premise AI Fund has set. If a technology is a general-purpose one that spreads across society the way electricity did, then it should create new uses simultaneously across several seemingly unrelated industries, not just one. When electricity was first introduced, factory production lines, household lighting, and hospital medical equipment each adopted it in their own different ways. A technology that changes only one industry is, by that measure, not so large a technology. The reason AI Fund builds companies across multiple industries is also part of testing this very hypothesis the organization has set for itself. Whether this portfolio spread was a deliberate strategy from the start, or simply the natural result of following the ideas of the founders who came through the door, cannot be determined from public materials.

The problem AI Aspire targets touches on what the industry commonly calls pilot purgatory. Many large enterprises follow a familiar path when they try to adopt AI. One department starts a pilot project with a small budget, the project looks successful in the conference room, and the people involved feel ready to move to the next stage. Yet that pilot project rarely spreads across the whole company. The reason often lies not in any shortcoming of the technology itself but in the organization's structure. What worked in one department does not fit another department's data systems or work processes, and exceptions that were overlooked during the pilot phase come back as obstacles at the company-wide rollout stage. Where Landing AI

focused on transforming a single factory or a single department, AI Aspire has positioned itself at the stage beyond the pilot, the stage of spreading across the whole organization. Its partnership with a major management consulting firm like Bain & Company can be understood in this light. Knowing AI technology is one kind of task; changing, together, the decision-making structure, budget allocation, and personnel evaluation criteria of an organization with thousands of employees is another kind entirely. The latter lies closer to the domain of a consulting firm that has long dealt with management organizations, while the former is the domain Ng has built up over a long time. The union of the two organizations can be seen as an attempt to bring both under a single advisory umbrella.

Joining Amazon's board is a different kind of position from the two organizations discussed above. At AI Fund and AI Aspire, Ng held positions directly involved in the day-to-day operation of the organization. A board is different. The board of a publicly traded company is not a seat that handles the company's daily business directly but one that oversees the major decisions management makes and, on shareholders' behalf, approves or blocks broad direction. A director is closer to an overseer bearing a fiduciary duty to shareholders than an employee working for the company. As generative AI rose as a topic across many industries in the 2020s, not just Amazon but many publicly traded companies showed a trend of wanting a board member who deeply understood AI. To judge the decisions a board makes — which businesses to put capital into, which risks are worth taking, whether the AI-related business plans management presents are realistic — the directors themselves need some grasp of the technology's limits and possibilities. The passage in Amazon's statement recruiting Ng, saying his experience was needed to help the board understand the opportunities and challenges AI opens up, can be read against this backdrop. The fact that his history as a researcher writing papers, running organizations at Google Brain and Baidu, and founding Coursera and Landing AI was cited together points to the same context. What Amazon needed was not someone who knew only the technology but someone who had broadly experienced the many layers of problems that arise when technology is actually carried over into business.

A board member's work does not show much on the surface. Board meetings are not public, and there is no way to confirm from public materials what discussions took place in those meetings, what questions Ng raised, or which items he opposed or supported. There is no basis at this point to judge what actual influence he has had as a director. What the fact of his taking this seat does show, however, is that the scope of his activity has widened beyond startups and mid-sized companies into the governance structure of one of the world's largest technology companies.

Set alongside the earlier chapters, Ng's work as a mentor reveals one continuous thread. At Coursera and DeepLearning.AI, he made it so that anyone who wanted to learn artificial intelligence did not need to cross the threshold of a classroom. On AI Fund's Insights channel,

he has made it so that anyone who wants to start a company does not need to pay for expensive advisory fees. Posts on how to write a pitch deck, how to hire a first salesperson, and the technical tradeoffs involved in working with large language models are published free of charge, readable not only by founders who received direct investment from AI Fund but by anyone outside it as well. The way of sharing knowledge through lectures has simply taken the form of practical guidance here; the direction of lowering barriers has not changed much. The case of inviting Pandora founder Tim Westergren for a conversation also fits within this pattern. It was an occasion for a founder who had spent a long time building a company to pass on the trial and error he went through to the next generation, not through a classroom but through conversation.

The program called Engineer in Residence is also an extension of this mentorship. This type of program, commonly run by venture studios, keeps an engineer not yet attached to any specific company inside the organization, moving between several early-stage projects to solve technical problems together. It is a device for letting an engineer build a relationship with the organization even before a company is formed. The Builder events held in Mountain View, likewise, function less as fundraising occasions than as gatherings where people who have not yet founded a company meet one another and test out ideas. The goal Ng has set forth at these events — building great companies that advance humanity — connects with language he has repeated for a long time. However, exactly which founder was told this line, in what situation, is not confirmed in public materials.

The written statement he submitted to the U.S. Senate's AI Insight Forum is also worth noting. This forum was a series of closed-door briefing-style sessions held several times in 2023, bringing together technology company executives, researchers, and civil society representatives to discuss the direction of AI regulation. It is not so unusual for a figure with a research background to submit written opinions to such a forum, but the fact that Ng was called in likely has to do with his background spanning academia, industry, and education. That he was chosen as a voice policymakers might consult means his remarks have now reached beyond a single company or a single course, into discussions at the national level. Time magazine's inclusion of him on its 2023 list of the 100 most influential people in AI is one indicator of how this broad range of activity was read from the outside.

The three organizations covered in this chapter — AI Fund, AI Aspire, and the Amazon board — each operate at a different level. AI Fund is a place for building companies that do not yet exist; AI Aspire is a place for changing the internal structure of large enterprises that are already established; the Amazon board is a place for overseeing the major decisions made by a company of global scale. In all three seats, Ng stands not as someone who directly builds technology but as someone who helps technology spread into the world through the forms of organization, capital, and policy. How much actual change these activities have produced, how

many of the portfolio companies have survived, and what positions he has voiced on the board cannot be known from public materials available at this point. What can be confirmed is only the fact that he holds all three positions at once, and the goals those positions aim at.

To fully understand the approach Ng took at AI Fund, it helps to compare it with two other approaches already established in the startup ecosystem. One is the accelerator, which gathers several teams together for a short period, offers advice on a fixed schedule, and puts in a small amount of money in exchange for only a small stake. The other is traditional venture capital, which appears after a founder has already started a company and built a product with some shape, provides capital in exchange for equity, but does not get deeply involved in running the company. The venture studio model AI Fund chose differs from both. It gets involved before the company is even born, refining the idea, assembling the team, and helping set the product direction together. This difference is reflected directly in the equity structure as well. The stake an accelerator takes is not large, and the stake traditional venture capital takes is priced against value the founder has already built up. Because a venture studio works together from before the company existed, it commonly takes a structure in which the founder and the studio split equity more evenly. Under this structure, the share of equity a founder ends up holding may shrink, but the time and risk involved in starting the company shrink correspondingly.

It is also worth noting that Ng uses a different title at AI Fund than at AI Aspire. At AI Fund he uses the title managing general partner. This title refers to someone who manages the operation of an investment fund, and it comes paired with a structure in which, if the equity value of companies the fund invested in rises, a portion of that gain is distributed back to them. At AI Aspire he uses the title managing partner. This title is one commonly used at professional service organizations that provide services in exchange for advisory fees. The difference between the two titles shows that the two organizations earn revenue in different ways. AI Fund profits from building companies and seeing their value rise, while AI Aspire earns fees for providing advisory services to large enterprises. The same person is involved in the spread of artificial intelligence through two different methods at once.

The makeup of the limited partners who funded AI Fund is also worth revisiting. New Enterprise Associates and Sequoia Capital are themselves venture capital firms that invest directly in startups. It is not unusual for such institutions to also put money as limited partners into a fund run by someone else, beyond the funds they manage themselves. Large venture capital firms sometimes choose to entrust capital to an organization skilled at building companies together with founders at the earliest stage, and share in the resulting gains. The calculation behind why capital flowed into AI Fund can be understood in this light. However, what internal review each institution went through in deciding to invest cannot be confirmed from public materials.

The program called Engineer in Residence also belongs to a device widely used in the venture studio industry. Keeping an engineer not yet attached to a specific company inside the studio, moving between several early-stage projects to solve technical problems, is a device meant to resolve one common difficulty a studio runs into when building a company. Even with a good idea, if there is no engineer to turn that idea into code, the company cannot get started. When a studio secures engineers in advance, the speed of pairing ideas with people increases. The fact that AI Fund runs this program shows that the organization treats bringing people together, not just capital, as part of its own job.

The trend of wanting a board member who understands artificial intelligence was not confined to Amazon alone. Heading into the mid-2020s, many publicly traded companies showed a movement toward placing someone with deep understanding of AI technology among their board members, to help management's judgment. Within this trend, Ng's joining Amazon's board can be read not only as an event specific to him personally but also as one instance of a broader change taking place across technology-company governance structures. Because a board is a place that approves or blocks a company's business plans, whether directors have the ability to gauge the feasibility and risk of a particular technology matters to shareholders as well. This need is also part of the background behind Amazon's statement laying out his experience item by item.

That the posts AI Fund publishes on its Insights channel are free of charge overlaps with the approach Ng maintained at Coursera and DeepLearning.AI. The scope of the audience, however, differs. Coursera's courses were aimed at anyone who wanted to learn artificial intelligence, while AI Fund's practical guidance is aimed only at those who have decided to start, or are preparing to start, a company. As the audience narrows, the content becomes more specific. Posts on how to write a pitch deck or how to hire a first salesperson deal with a different layer of knowledge than an introductory course that has to explain what artificial intelligence even is. This difference shows that Ng has kept adjusting the way he lowers the barrier to education according to the audience.

Setting the three roles covered in this chapter side by side reveals that the nature of Ng's work has changed. At Google Brain, he designed neural networks himself; at Coursera, he recorded lectures himself. At AI Fund, AI Aspire, and the Amazon board, he does not write code or design products. The knowledge and experience he has built up are used only as material to help other people's judgment. This change is closer to meaning that the channels through which his influence spreads have multiplied, not that his influence has diminished. There is a limit to what one person can build directly, but that kind of limit applies relatively less to the work of handing material for judgment to many founders, many companies, and many policymakers. The many companies AI Fund has nurtured, the many large enterprises AI Aspire has advised, and the major decisions the Amazon board handles each exceed a scale

that one person, Ng, could manage alone. This is likely also the reason he has been able to hold all three roles at once. He is no longer someone who solves one problem within one organization, but someone who hands frameworks for judgment to many organizations.

Realism in the Face of AGI Fever

In 2026, the artificial intelligence industry is throwing out wildly different numbers for when artificial general intelligence, or AGI, will arrive. Elon Musk has staked a claim of one to two years. Anthropic's Dario Amodei foresees something even sooner, roughly two to three years, for reaching human-level intelligence. Google DeepMind's Demis Hassabis speaks of a range between five and ten years. Each has reasons behind the number. Model performance has climbed visibly year after year, and benchmark scores keep pushing past the limits of the previous generation. Andrew Ng, however, does not stand in this line. He states flatly that true AGI is still decades away. Among colleagues who talk in terms of a few years or at most a decade, the reason he insists on a different number lies less in the prediction itself than in how the word behind that prediction is being used.

What Ng objects to is the fact that the word AGI has lost its academic definition. Originally the term referred to a system that could broadly perform the intellectual work a human can do, in contrast to narrow AI that excels at one specific task. Now, though, it gets used evenly across fundraising announcements, marketing copy, and policy debate rhetoric, and its original meaning has blurred. Ng explains the situation through an analogy with color. What would happen if people started calling red blue, calling yellow blue, and even calling green blue? Before long the word blue would point to nothing at all. Even if someone said blue, there would be no way to know what color they actually meant. His diagnosis is that the word AGI is now walking down this same road. As the label gets attached to systems of every level, the picture that comes to mind when people hear the word AGI has become different from person to person.

This confusion does not stay confined to academic seminars. Ng worries that a good many high school and college students are changing their majors because of the mistaken belief that AGI will soon eliminate every job. They are deciding their careers right now based on a technology that has not even arrived in the world yet. Corporate CEOs are no different. They make capital allocation decisions leaning on excessively optimistic forecasts, and when those decisions turn out wrong a few years later, the losses fall squarely on their organizations and employees. What Ng is truly worried about goes a step further. Once inflated expectations fail to hold and turn into public disappointment, a phase can arrive in which investment and attention drain away all at once. He calls this phase an AI winter. In fact, the history of artificial intelligence research has already seen two such winters. When expectations ran ahead and results failed to keep pace, budgets and talent moved to other fields, and research froze for a long stretch. Ng believes today's hype carries the risk of retracing those same steps.

To clear away this confusion of language, he proposes a single standard, which he calls the Turing AGI test. The Turing test that Alan Turing proposed long ago asked whether a machine

could fool a person through conversation. The new standard Ng has refined starts from a different place. An AI is given the exact same computer that a human remote worker actually uses — a browser, video-conferencing software, and a file system all installed just as they would be for a person. Then, over the course of several days, the AI is observed to see whether it can handle real, economically valuable work as capably as a human. A person acting as judge keeps throwing new tasks at it, trying to find its limits, and even after days of this pressure, the standard is met only if the AI shows no noticeably worse performance than a human would. By this yardstick, Ng's judgment is that no model today has yet crossed the threshold. Plausibly pulling off a few tasks in a demo video and holding up without faltering through days of unfamiliar demands in a real work environment are different orders of accomplishment.

Even as he clears away the hype surrounding the grand name AGI, Ng does not turn his back on the substantive changes that today's artificial intelligence is already producing. At the center of that change sits the term agentic AI. He is one of the people who has been building and refining this concept for some time now. What an agentic system does differs from the way a person asks a question and gets a single answer spat back. Just as a person handles work through multiple steps, AI too works through several stages on its own to complete a task. Ng points to four recurring patterns that appear when designing such systems: reflection, in which the system reviews and refines what it has produced; tool use, in which it calls on external tools such as search or a calculator; planning, in which a large goal is broken into smaller steps and sequenced; and multi-agent collaboration, in which multiple AIs, rather than a single one, each handle their own portion of a task and exchange results with one another. Combining these four patterns lets AI go beyond simple question answering to write and fix code, review documents for compliance with tariff regulations, read through complicated contracts to extract the essentials, or handle customer support and medical assistance work across multiple steps.

The trouble is that this term, too, has begun walking a path similar to AGI's. Ng admits that when he refined this concept and put it out into the world, he did not expect that before long the label agentic would be slapped onto every kind of product. In practice, the name got attached even to automation tools that simply call a few functions in a fixed sequence, and in this way the weight of the word grew light again inside marketing copy. There is one principle Ng repeatedly emphasizes at this point: a demonstration that looks impressive in a lab and a production environment that a company must rely on every day need to be judged by entirely different standards. When a researcher hands an AI several tools and lets it work freely at a task, remarkable scenes can result. But bringing this into an actual company changes the picture. A company demands the certainty that the same procedure will produce the correct result whether it is run ten thousand times or a hundred thousand times. Ng's diagnosis is that today's AI models still carry unpredictable corners when they must make judgment calls on their own, and the gap in trust is too wide to let them loose without any oversight. That is why,

on the industrial front lines, the chosen approach is to break complex work into many small steps, automate each step in a stable way, and stitch the results together. Step-by-step control, in other words, is a more reliable choice at this point in time than flashy autonomy.

Layered onto this is another paradox. When a large language model is handed too many tools and too much explanation all at once, its performance actually drops. As the list of tools grows longer, the amount of information — the context — the model has to take in at one time swells, and attention scatters away from what actually needs watching, leading the model to call the wrong function or lose sight of the original goal. Even the widely used Model Context Protocol, or MCP, servers are not immune to this problem. Lining up dozens of tools at once leaves the model struggling over which one to pick, and that confusion only grows as the number of tools increases. In the end, the conclusion Ng draws is that simply increasing the number of tools is not the answer in itself. He speaks highly of the skill involved in the development tool set, the so-called harness, that Anthropic built around Claude Code, along with the technique of structuring prompts carefully — and rates it as still playing a decisive role in real-world application. His view is that no less than the model's own performance, how that model is wrapped and in what order instructions are handed to it determines success or failure in practice.

Having coolly pinned down the limits of the technology in this way, Ng turns his attention directly to the question of jobs. On the fear that AI will sweep away human labor wholesale — the so-called end-of-jobs thesis — Ng, along with scholars such as his Stanford colleague Fei-Fei Li, shakes his head. Their shared view, looking back at the Industrial Revolution, is that new machines expanded the scale and manner of labor rather than eliminating it outright. In tackling this issue, Ng shifts his gaze away from the large unit of the job to the smaller unit inside it: the task. A single job is in fact made up of a great many individual tasks. Even one job title such as nurse mixes together caring for patients, keeping medical records, checking test results, and communicating with other medical staff. According to Ng's analysis, in most jobs the share of tasks that AI can actually take over falls somewhere around 30 to 40 percent. Turned around, this also means that the 60 to 70 percent that sustains that job still requires a human hand.

This number carries no small significance. Picture nurses and doctors buried under paperwork such as extensive medical record-keeping rather than the core task of caring for patients directly, and AI looks less like something taking these people's jobs away and more like a tool freeing them from draining, repetitive work. So the core sentence Ng offers comes down to this: people who use AI will end up replacing people who do not use AI. The point Ng stresses here is that the one doing the replacing is not the machine but the colleague who has skillfully brought that machine into their own work. It means that only those who reshape how they work to match each new tool as it appears can hold onto their place going forward. This is

not the first time such a thing has happened. The same fork in the road appeared when computers and spreadsheets entered the office. The fact that statistics exist showing that lifetime income fell and mortality actually rose among those who failed to adapt to that shift shows how, at every technological turning point, a gap in adaptation can turn into a gap in life outcomes.

That said, Ng is not saying that every job is safe. On this point he answers directly, without hedging. He does not hide the fact that job categories exist where AI really can take over nearly all of the work. Translators, voice actors, and call-center agents, he says, stand before the clearest threat right now. These jobs are the exception that falls well outside the 30-to-40-percent average mentioned earlier. Translating language, lending a voice, and answering fixed questions are areas close to what today's AI can already do at a level that rivals a person. Ng does not stop at acknowledging this fact — he moves straight on to the question of society's responsibility. His position is that just because someone's job has vanished due to technological progress does not mean it is acceptable for that person to be pushed out onto the street with no preparation at all. He says society has a moral obligation to look after such people. This is where his call for government and educational institutions to properly build out retraining programs and a social safety net comes from. Care for the people who bear the losses caused by technological progress, he argues, is not something to set aside separately — it must be handled as one matched pair with the progress itself.

Moving past this practical discussion surrounding jobs, Ng turns his attention to a debate raging in academia over the very nature of human intelligence. Yann LeCun and Demis Hassabis paint different pictures of the human brain. LeCun sees the human brain as the product of many highly specialized modules coming together — one part handling language, another processing vision, another regulating movement. Hassabis, by contrast, says the human brain is a single, highly adaptable general-purpose learner. Ng judges that there is actually no contradiction between these two views. The true wonder of the brain he points to lies precisely in plasticity — the capacity to learn new things in the first place. A person who holds a doctorate in mathematics has a brain deeply specialized in that field today, he explains, but had that same person been trained differently in a different environment, they might just as well have become a chess master or an outstanding tennis player. The point is that humans are born with a general-purpose learning engine from the start, and then shape themselves in a particular direction through the experiences that follow. Ng sees today's AI walking a similar path. A single model's general-purpose potential is, in practice, taking concrete shape as narrow applications solving specific problems in fields as different as medicine, manufacturing, and finance.

This optimistic view leads into another debate surrounding AGI: the debate over existential risk to humanity. Figures such as Eliezer Yudkowsky warn that the moment AGI is created, all

of humanity will be placed in danger. Ng does not agree with this claim. He points out that Yudkowsky's logic is so circular that it is hard to know even where to begin refuting it. His own way of seeing the world starts from a different place. He first calls to mind the fact that if the advance of technology is slowed now, millions more people will go on losing their lives to cancer, aging, and various diseases in the years ahead. If, on the other hand, the advance of AI is pushed forward, he believes a path opens to save more lives, create new wealth, and lift people out of poverty. Ng's realism, in the end, holds two attitudes within a single frame: a clear-eyed eye that coolly sorts out the bubble now heating up the market, and a belief that the benefit this technology will bring to humanity still far outweighs the losses it causes. Clearing away expectations inflated by exaggerated words and continuing to push forward tools that actually improve people's lives were, to him, not two different directions but two expressions arising from one and the same attitude.

Looking a bit further into the circumstances of various jobs shows that the 30-percent figure Ng cites does not apply equally across every industry. Take legal work as an example. Among the tasks a lawyer handles, language models can take over a considerable share of case-law research and catching errors in draft contracts. But meeting with clients to hear out their circumstances, persuading judges and juries in the courtroom, and reading the other side's expressions across the negotiating table to work out terms remain, still, in human hands. The manufacturing floor shows a similar pattern. Inspection work that picks out defective products using cameras is something vision AI does faster and more accurately than a person, but placing new equipment, watching over worker safety, and improvising responses to sudden breakdowns are hard for a machine to take over. This is why Ng frames the problem in terms of tasks rather than jobs. Lumping together all the different work bundled under a single job title makes it easy to reach a hasty conclusion, but breaking that work down piece by piece makes far clearer where artificial intelligence can dig in and where a human still needs to hold the line.

This view carries straight over into education as well. Ng says what universities and vocational training institutions ought to teach students is not how to operate a handful of particular software programs, but the judgment to decide which part of the work in front of them to hand off to a machine and which part to keep taking responsibility for themselves. This judgment is needed just as much by older workers as by young students. If a new employee hands off repetitive paperwork to AI from their very first year and pours that freed-up time into building relationships with clients instead, that person's career will diverge sharply, a few years down the line, from a colleague in the same role who kept holding onto nothing but paperwork. What Ng means by retraining, in this sense, is closer to building the habit of reshaping how one works altogether than to picking up one new programming language.

The change brought on by agentic workflows is also reaching into the structure of corporate organizations. There are more and more cases where work that a single person used to handle from first step through middle step to final step is now divided among different AI agents, and the person shifts to a role of checking the results in between and refining direction. Ng does not see this reshaping flowing only in the direction of shrinking people's place. Rather, he explains, as the range of work a single person can oversee widens, teams emerge that handle more work with fewer people than before, and inside those teams the role demanded of a person shifts from someone who directly carries out the work to someone who reviews and coordinates the work of multiple agents. In such an organization, alongside the ability to operate AI, the judgment to pinpoint exactly what an AI's output got wrong rises up as a new core competency.

Ng maintains a cautious stance regarding the pace of this change as well. A demo shown at a Silicon Valley launch event looks like a feature completed within a few days, but it takes far longer for that feature to actually take root inside a company's real work procedures. The time it takes for a single piece of new software to be validated against a hospital's or a bank's operating rules, and for the staff responsible to trust and actually use that tool, is always slower than the pace at which the technology itself advances. Ng points out that people commonly misjudge this gap. Just because an AI model's performance leaped forward dramatically within a single year does not mean the pace of actually deploying that model into real work speeds up at the same rate. In fact, the wider this gap grows, the more a mismatch opens up between technological advance and change on the ground, and hasty attempts to close that mismatch become, in his view, one more thing inflating the very bubble he worries about.

What he repeatedly stresses in venues such as the World Economic Forum in Davos is no different. Speaking to policymakers and business executives, he says both extremes surrounding the discussion of artificial intelligence need to be guarded against together. One extreme is baseless optimism that AI will take over all human work within a few years; the other is baseless fear that AI will ultimately drive humanity to ruin. His point is that both attitudes end up missing what is actually happening right now — the quiet shift in which specific tasks get automated one by one and human roles get newly rewoven in between. This quiet shift, Ng says, is exactly what governments and companies should actually be grabbing onto as they draft policy and budgets. Deciding how to help people who are losing jobs right now in call centers, translation agencies, and factories, he means, is more urgent than setting up laws to prepare for a superintelligence that has not yet arrived.

This attitude also connects with the advice he repeatedly offers students and practitioners. Rather than pouring effort into predicting how far AI will advance within the next few years, he urges people to directly experiment with what can be built using the tools already in hand.

Only someone who has confirmed a tool's limits firsthand can pinpoint exactly where that tool falls short — that is, exactly where a human still needs to fill the gap. To Ng, artificial intelligence is not some mysterious being destined one day to surpass humans, but a tool whose performance and limits are on display together right at this very moment. The task of the person handling that tool lies in finding what the tool cannot do and pouring one's own time into that work. This attitude is the axis that holds up his realism, and at the same time the stance he has kept as an educator all along.

Picking up this question from a slightly different angle, Fei-Fei Li addresses the same issue. In one interview, she said that ten years from now only two kinds of workers will remain: people who direct AI in its work and review the results, and people who move according to what AI instructs. This means the labor market will split along that line. Where the 30-to-40-percent figure Ng pinpointed earlier finely divides which tasks within a single job shift over to machines, this statement is closer to asking how that outcome will rearrange people's standing relative to one another. Even within the same hospital, the same law firm, the same factory, a clear line is beginning to form between the position that skillfully wields AI tools and sets direction, and the position that checks and lends a hand to what those tools have produced. Ng does not accept this division as a matter of fate. The need for retraining that he has stressed again and again comes from a belief that which side of this fork a person ends up on is a matter of training rather than innate talent. Standing in the position of giving direction requires firsthand experience of what AI is good at and what it misses, he judges, and this experience is the kind of thing anyone can build up regardless of age or educational background.

This reshaping was also discussed under a different name at the World Economic Forum gathering in Davos. Broadcasts called it the Great AI Reshuffling — the organizational overhaul touched off by artificial intelligence. The corporate promotion ladder has long been structured so that new hires take on repetitive hands-on work, build up experience through it, and climb a rung at a time using that experience as a foothold. A junior lawyer reviews hundreds of contracts and develops an eye for spotting traps in clauses; a junior analyst tallies financial statements by hand and picks up a feel for reading the relationships among numbers. But this very repetitive work at the entry level is precisely the area today's language models can take over first, and most cheaply. This is exactly where Ng's worry diverges. When repetitive work disappears, things look easier for new hires in the short run, but the very opportunity to build the judgment that could only be gained through that repetition shrinks. If the lower rungs of the ladder pass over to artificial intelligence, where the people who will climb to the upper rungs a few years later will be trained becomes a new homework problem. The discussion that took place in Davos treated this as a challenge for companies to redesign the way they hire new employees in the first place — assigning, from the start, the work of verifying and correcting what AI has produced instead of repetitive tasks, and building new training paths

that let judgment be developed through that process.

In the setting the World Economic Forum arranged, Ng does not confine this issue to individual companies' personnel policies but expands it into a matter of how society as a whole converses. His reason for bringing up the phrase honest conversation at this point is his diagnosis that today's public discourse is locked into two extreme languages. One is the language that treats AI as a being that will either save or destroy humanity within a few years; the other is a silence in which, crushed under this grand narrative, the small changes actually happening right now do not even register as news. What policymakers need, Ng says, is a language that sits somewhere between these two extremes and plainly points out what tasks are being automated right now and who is actually losing income as a result. Only once such a language takes hold, he argues, can governments make timely, concrete decisions — such as where to allocate retraining budgets first or how to adjust unemployment benefit standards. His repeated argument that budgeting to respond to the workforce adjustments already happening in call centers and translation agencies takes priority over legislation to prepare for the arrival of superintelligence is the practical conclusion that follows from this principle of honest conversation.

One point Ng makes with particular force in this conversation is the fact that the debate surrounding artificial intelligence is unfolding only within a small handful of giant companies and a small handful of wealthy countries. Drawing on his experience working at AI Fund and Landing AI, he has repeatedly tried to show that a single manufacturer or a single hospital can solve problems with its own data even without expensive supercomputers or massive datasets. What he asks of policymakers in venues such as Davos is no different. The center of gravity in the discussion, he argues, needs to shift from the competitive dynamics among the handful of companies capable of building giant models to the far greater number of companies and workers who take those models and apply them to the specific problems of their own industries. Seen from this angle, his calls surrounding retraining and a social safety net are not a proposal to slow the pace of AI's advance, but closer to a demand that the benefits and burdens that advance produces not be funneled solely toward a narrow slice of people. Believing that the progress of technology itself cannot and need not be stopped, while still trying to match the speed at which the fruits of that progress spread to the speed at which people can adapt — this attitude forms the backbone of the message he repeated in Davos.

Looking at the names that appear in this debate offers a clue as to why such different numbers keep coming up. Elon Musk, Dario Amodei, and Demis Hassabis are all people leading organizations that directly build giant models on the front line. They stand in a position to watch, ahead of almost anyone else, models growing newer month by month inside their own labs. Their predictions, as a result, often rest on the premise that the performance curve will keep climbing at the same slope it has now. Ng questions the optimism that comes from

this vantage point. The speed at which benchmark scores rise inside a lab and the speed at which that performance actually settles into a hospital's medical-record system or a bank's loan-review procedure run on entirely different clocks — this is the point he repeatedly makes. A benchmark is a test of solving a fixed problem set, while the real world is a place where new exceptions pop up every day.

The errors CEOs make in allocating capital also stem from this time lag. Some companies, believing they can shift their entire customer-support workforce over to AI within a few years, cut hiring first. Once the system is actually put in place, though, there remains the problem that even if it delivers smooth answers nine times out of ten, the tenth time brings an incident of giving a completely wrong instruction. This kind of error rate goes unnoticed on a demo stage, but in a call center handling tens of thousands of inquiries a day, it piles up into a daily stream of repeated complaints. In the end, the company finds itself having to rehire the staff it cut, or belatedly redesign a procedure in which people and machines work together. What Ng calls a capital allocation mistake is a decision that arises, in this way, from viewing the gap between performance and reliability as far narrower than it actually is.

The reason the Turing AGI test insists on a multi-day trial in the first place also ties back to this gap in trust. Producing a plausible-sounding answer to a single question and holding up through an actual work week running from Monday to Friday demand different things. Fresh questions get thrown at it every day: does the approach used on Tuesday still hold for a slightly different request that comes in on Wednesday; does it notice when a meeting schedule suddenly changes and adjust the follow-up work on its own; does it ask for clarification when the judge deliberately gives an ambiguous instruction. Passing such a test requires not one brilliant answer but consistency that does not waver across several days. The very absence of that consistency is Ng's grounds for stating flatly that none of today's models has cleared this bar.

His caution surrounding agentic workflows springs from the same root. The four patterns of reflection, tool use, planning, and multi-agent collaboration each unlock powerful capabilities, but the more of these layers are stacked on top of one another, the more channels also open up through which the system can veer off course somewhere in the middle. If the goal is set wrong at the planning stage, every tool call that follows becomes futile from the start; and in a structure where multiple agents collaborate with one another, one agent's mistaken judgment can be carried over unchanged by the next agent, compounding the error. This is why the teams Ng has worked with on the industrial front lines choose, rather than turning all four patterns loose freely at once, to leave a checkpoint at every step where a human can verify what is happening. It means giving up a bit of freedom in exchange for keeping a structure that can pinpoint exactly where something went wrong.

The paradox that more tools mean worse performance also connects to this same design principle. Just as a person handed dozens of rulebooks all at once on their first day at a new job is likely to miss the important rules, a language model, too, struggles more to pick out what is actually needed at this particular moment when the list of tools it must process at once grows longer. That is why teams that deliver results in practice choose to hand the model only the handful of tools truly needed for the task at hand, rather than handing it every tool all at once. This is also why Ng keeps a close eye on the harness Anthropic has built up around Claude Code. Rather than entrusting the model with everything at once, this harness is built to select only the information and tools needed at this particular step and pass the result on to the next step. What he has confirmed again and again is that this kind of design — selecting and handing over information — determines success or failure in practice no less than the model's own raw performance.

A similar principle repeats itself in the discussion surrounding jobs as well. The figure of 30 to 40 percent largely overlaps with the share, within a single job, of standardized work — that is, work that follows a fixed procedure and repeats. Organizing documents, transcribing records, and giving a fixed answer to a fixed question are all in a form language models handle well, while making impromptu judgment calls in unfamiliar situations and reading a person's expression and tone to build trust remain, still, in human hands. The reason translators, voice actors, and call-center agents face a notably higher risk is that most of the work these jobs involve is close to standardized tasks — rendering a fixed language into another fixed language, lending a voice to a fixed script, or turning a fixed question into a fixed answer. The reason Ng singles out and names these three occupations separately is so as not to hide the fact that the 30-percent average applied to the rest of the jobs does not hold for them.

The story surrounding the brain's plasticity is also tied to this same distinction between the standardized and the non-standardized. When Ng says that a mathematics PhD, trained differently, might have become a chess master or a tennis player, he means to show that human intelligence is not fixed from the outset to one narrow use. Today's AI models, too, started out with a general-purpose capacity for handling language, but as they are placed into the fields of medicine, manufacturing, and finance, they go through a process of being refined to fit the language and procedures of each industry. This process resembles the one a person goes through after graduating college and spending a few years at a particular job growing accustomed to the ways of that industry. Ng's long-standing observation of this kind of resemblance is the backdrop against which he explains today's AI not as some mysterious leap but as a familiar process of learning.

The existential-risk debate raised by Yudkowsky, too, can be read afresh within this same context. Ng calls that argument circular because its logic already lays down as a premise the conclusion that AI is dangerous, and then goes on to fetch supporting evidence after the fact

to prop up that premise. Ng's own judgment, by contrast, starts from the numbers right in front of him. He first counts the number of people who die each year from cancer and aging — a number that grows larger every time the technology is delayed — and then weighs, side by side, AI's potential to reduce that number against its potential to create new risks. In this weighing, he places more weight on the side where the benefit outweighs the risk. His call for honest conversation in Davos was, in the end, a request to redo this weighing not with the language of abstract fear or vague salvation, but on the basis of numbers that can be counted right now and changes that are actually happening right now.

Support This AI Library Book

If this book was useful, you can support future translations, public editions, and open AI knowledge projects through PayPal.



PayPal

[**https://paypal.me/kimkjj**](https://paypal.me/kimkjj)

Scan the QR code or open the link above.