Nurture AI like you would a child.

For years now, even before ChatGPT placed artificial intelligence (AI) firmly at the forefront of the public imagination, AI has been slowly taking hold across industries – from medicine to aerospace. However, the technology hasn’t quite lived up to its potential. Not even close.

A recent study found that only 11 percent of firms using AI have gained financial benefits. Even technology giants have struggled. IBM’s US$20 billion diagnostic AI system Watson Health diagnosed cancer more accurately than doctors in laboratory experiments but flopped in the field. It became a commercial and reputational disaster for the storied American MNC.

The failure could hardly be blamed on a lack of technical expertise, for IBM tasked an army of engineers to work on Watson Health. Our extensive research on the challenges of developing AI in various commercial settings points to a surprising cause: Watson Health was developed and brought to market in a way that works well for traditional IT but not AI. This is due to a
fundamental difference between conventional software and AI: While the former processes data, AI continually learns from the data and becomes better over time, transcending even its intended capabilities if nurtured properly.

We argue that practices analogous to the best parenting styles can accelerate AI development. We prescribe an AI development approach, detailed below, that is based on nurturing and learning, which we have recognised as a key ingredient for success in more than 200 AI projects for industrial and other customers.

1. Deploy early and learn from mistakes

Children do not learn to cycle by watching an educational video but by clambering onto a bike and stepping on the pedals, learning valuable lessons from each painful fall – before long, the magic happens.

The same logic applies to AI. Many companies such as IBM think they should collect vast amounts of data to perfect the algorithms before deployment. This is misguided. Putting AI to work in the real world, rather than sequestering it in controlled environments, helps generate more data that in turn feeds back into the development process.

Although early deployment is inherently riskier, it also initiates a continuous feedback loop through which the algorithm is enriched by new data. Further, it is important that the data stems from both standard and difficult or atypical situations that, taken together, support comprehensive AI development.

ChatGPT is a great example. The chatbot was released to the public by OpenAI in November 2022 while still wet behind the ears, albeit more for reasons concerned with getting ahead of the competition. In any case, the gamble worked: Not only has ChatGPT become a worldwide phenomenon, leaving the likes of Google’s Bard scrambling, its early launch also garnered millions of users and generated vast amounts of data for OpenAI to push out GPT-4, an improved version of the bot, only months later.

Another example is Grammarly. The finessing of its writing assistance system with the help of user feedback showcases the power of continuous AI improvement and adaptation, particularly in the complex and context-sensitive realm of languages.
Apodigi, a frontrunner in the digitalisation of the pharmacy business, launched an AI-assisted pharmacy app in June 2020 that can be described as learning on the job. Called Treet, the app proposes medication based on doctors’ prescriptions, which a pharmacist then reviews and tweaks. The pharmacist’s responses coalesce into a stream of continuous feedback that fine-tunes the algorithm and contributes to better recommendations that address the complexities of each and every patient’s needs and preferences.

By comparison, IBM developed and tested Watson Health extensively in the laboratory and pushed out the diagnostic tool to market without incorporating continuous learning from real-world data. This traditional build-test-deploy process proved inadequate for training AI.

2. Develop safety mechanisms

Safety mechanisms that protect consumers and safeguard reputations are essential in AI development. Simulator environments such as AILiveSim allow for full-scale AI systems to be safely and comprehensively tested before their deployment in the real world.

Meanwhile, Tesla runs new versions of its self-driving software in the background while a human drives the car. Decisions made by the software, such as turning the steering wheel, are compared to the driver’s decisions. Any significant deviation or unusual decisions are analysed, and the AI retrained if required.

AI developed for creative applications arguably needs stronger guardrails. Analogous to children mixing with bad company and learning undesirable habits, AI could be exposed to training data that are riddled with biases and discriminatory content.

To preempt this, OpenAI, for one, employs an approach called adversarial training to train its AI model to not be fooled by rogue inputs from attackers. This method involves exposing chatbots to adversarial content that threatens to overcome the bot’s standard constraints, enabling it to recognise rogue content and avoid falling for it in the future.

3. Capture user behaviour

In the ideal AI development cycle, developers log all user reactions and behaviour to feed further development of the algorithm without questioning the accuracy or value of a recommendation or prediction. The Netflix AI
content recommender, for example, simply notes whether a user watches the recommended content and the viewing duration.

Kemira, a company that specialises in sustainable chemical solutions for water-intensive industries such as paper and pulp production, developed a machine learning-based system to detect production problems before they happen and give recommendations to preempt breakdowns. The AI system learns from responses to make better recommendations going forward.

The developers of Watson Health could have achieved better outcomes if they had subscribed to this principle. Rather than programing the algorithm to ask doctors for their evaluation of the AI-generated recommendation, they could have trained the system to simply record doctors’ prescriptions. Integrating Watson Health into patient information systems would also have immersed it in a feedback loop for continuous training based on actual cases and patient outcomes.

User feedback provides excellent training data for vertical applications with a specific focus. Jasper, for one, is an AI-powered content generator that learns from users’ modifications to its proposed texts.

User behaviour could be converted into feedback at all stages of the AI’s learning process, which is composed of three parts: creating the teaching material; teaching; and collecting feedback on performance. First, developers collect labelled data to train the AI; the AI’s performance is then compared against desired outcomes or performance metrics; finally, feedback is collected that goes back into the training process, which repeats itself.

Data, especially labelled data, has become a crucial asset for AI companies. Rather than hiring humans to label data, developers should think of ways to automate the process as much as possible. For example, connecting the vehicle front camera feed with the steering wheel can automatically create labels for winding roads and feed into AI models learning to drive a car on complicated routes.

In fact, developers should deploy many automated data collectors and design explicit feedback loops for learning at scale. In the above example of driving assistance development, many vehicles can cover a wider variety of situations than just a few. A vehicle cutting in front of a Tesla triggers a video
upload from the previous few seconds preceding the event. The system feeds the footage into Tesla’s deep neural network that learns the various signals, such as a gradual movement towards the lane divider, that predict the cut-in and take appropriate action, such as by slowing down.

In contrast, traditional car companies are often mired in a fixed mindset, developing and deploying driving assistance software with little automated feedback collection or data updating.

4. Design for continuous learning at scale

Just as children do not stay in kindergarten forever, the training methodology for AI should be continuously upgraded. But all too often, AI developers focus on the latest developments in AI algorithms and individual use cases rather than engineering the system to cover a large number of use cases and data streams.

Kemira's machine learning-based system constantly accumulates insights into the root causes of potential instabilities while generating actionable risk-mitigating recommendations for paper machine operators. To ensure scalability, the system is cloud-based and uses MLOps for model retraining governance and to enable expansion to more use cases.

A crucial element of designing for learning at scale is a system architecture that collects feedback automatically, provides frequent updates for a large number of AI models and generates simulated training data. Suur-Savon Sähkö, a Finnish energy company, developed an AI forecasting method that learns from historical and real-time consumption data to improve the efficiency of energy production and the accuracy of heating supply temperature prediction by more than 50 percent.

Going one step further, companies can develop a simulation environment that generates synthetic data and allows for faster development cycles. For example, Tesla captures data from its fleet of cars to feed a simulator that simulates complex traffic environments, resulting in new synthetic training data.

AllLiveSim has developed a parametric, domain-agnostic simulation environment that supports machine applications in automotives, autonomous ships and autonomous mining. The simulator enables companies to build prototypes and verify concepts; create synthetic data
sets to train AI systems; debug and optimise algorithms; and test and validate products. It speeds up the development of machine-learning systems by capturing data and testing real-world cases that rarely occur.

To sum up, organisations that switch to a growth mindset and adopt the continuous learning methods described above are more likely to create AI solutions fit for a fast-evolving world. By nurturing algorithms with a constant stream of data and feedback, organisations can ensure that their products and services remain nimble, safe and relevant.

This article originally appeared on Fortune.com.

Find article at
https://knowledge.insead.edu/strategy/developing-effective-and-safe-ai-growth-mindset

About the author(s)

Tero Ojanperä is the co-founder and executive chairman of Silo AI, one of Europe's largest private AI labs.

Timo Vuori is an Assistant Professor of Strategic Management at Aalto University, Finland.

Quy Huy is the Solvay Chaired Professor of Technological Innovation and a Professor of Strategic Management at INSEAD. He is also a director of the Strategy Execution Programme, part of INSEAD’s suite of Executive Education programmes.

About the series

AI: Disruption and Adaptation

Delve deeper into developments in artificial intelligence, especially the disruptions across value chains. This series examines AI’s impact on a range of sectors, including business consulting, education and the media. It also sizes up the regulatory and ethical questions tied to this game-changing technology.