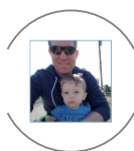


Recommended by you and 82 others



Robbie Allen

Follow

Startup founder turned Ph.D Student @UNCCS focused on AI, Machine Learning & Natural Langu...
Jun 13 · 17 min read

Why Artificial Intelligence is Different from Previous Technology Waves



Packed house before my talk at the first O'Reilly Artificial Intelligence Conference in Sept 2016. I thought we achieved Peak AI Hype then, but it's only continued to grow.

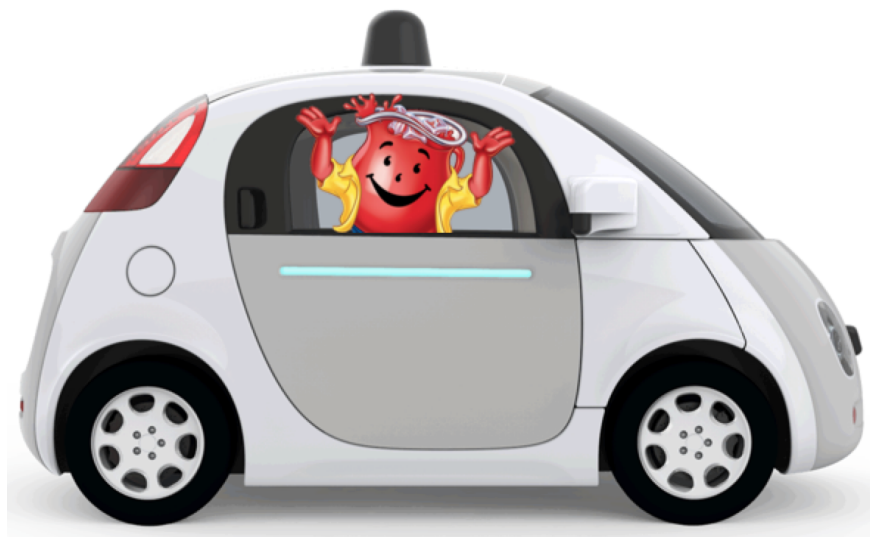
I've been around computing since my older brother got a Commodore 64 for Christmas in 1983. I took my first "business machines" class in high school in 1991, attended my first computer science class in 1994 (learning Pascal), and moved to Silicon Valley in 1997 after Cisco converted my internship into a permanent position. I worked in Cisco's IT department for several years before moving to their engineering group where I designed networking protocols. I went to grad school at MIT in 2004 where I met the founders of several companies in Y Combinator's first couple of batches and worked on Hubspot before it was Hubspot. After writing several books for O'Reilly and attending the first O'Reilly Web 2.0 and MIT Sloan Sports Analytics conferences, I started a "Web 2.0 for Sports" company called StatSheet.com in 2007, which in 2010 pivoted into the first Natural Language Generation (NLG) company called Automated Insights. I recently stepped back at Ai to become a Ph.D. student at UNC studying Artificial Intelligence.

All of that to say I've had a bird's eye view to watch the incredible innovation that's occurred over the past 30 years in technology. I've been lucky to be in the right place at the right time.

I've also seen my share of technology fads—I want my WebTV. Being a technology geek, I have a predisposition for shiny objects, and after managing over a hundred programmers in my career, it seems to be a

trait many developers share. Being older now and having seen a variety of technology waves come and go (including one called Wave), it's a little easier to apply pattern matching to the latest new thing that's supposedly going to revolutionize our lives. I've become more skeptical of most new technologies as I see people get enamored with silver bullet thinking.

While I believe Artificial Intelligence is overhyped in many ways, I'm extremely bullish about its potential even when you factor in all the hype. There is a unique combination of factors that will make AI the most impactful technology in our lifetime. Yes, I have drunk the AI kool-aid, but I think it is for good reason.



Mr. Kool-aid Man in a Google Self-Driving Car

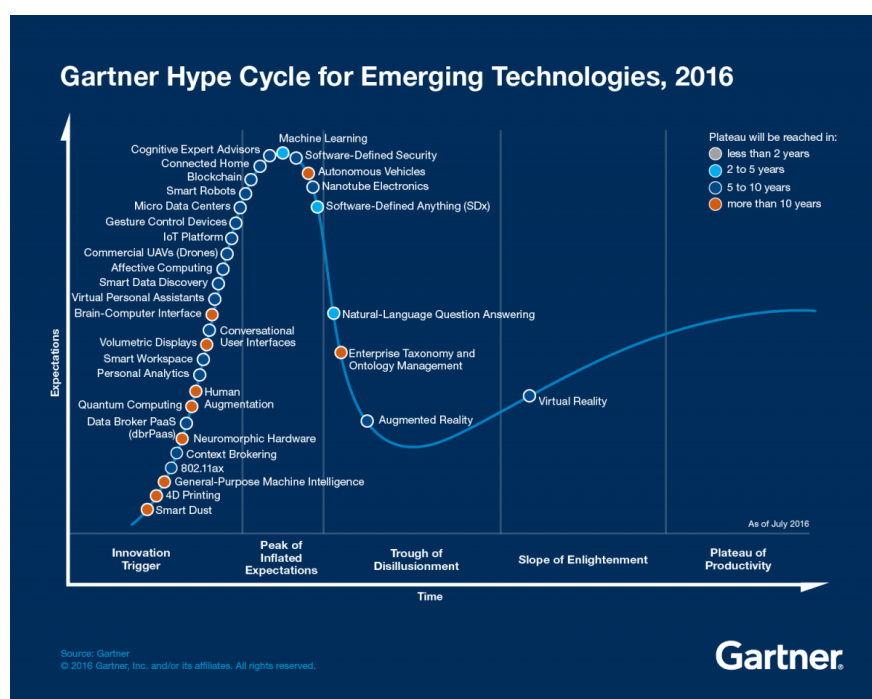
Selling AI “before it was cool” was not cool

As Artificial Intelligence started to make a comeback again in 2011, I was cautiously optimistic. I started a company in 2010 with the initials AI (for Automated Insights) before AI was cool. My intuition was that it would be a slog for a couple years, but if Ray Kurzweil was even partially accurate about the rate at which AI would happen, positioning a company in the AI space in the early 2010's would end up working out—it did.

In 2012, trying to sell AI as a capability was difficult to say the least. Our company built a solution that automated the quantitative analysis and reporting process. That is, we automated what had only been done manually before with handwritten, data analysis reports. Since then we've automated everything from fantasy football recaps for

Yahoo! to sales reports for one of the largest insurance companies. We automated earnings reports for the Associated Press and personalized workout recaps for bodybuilding.com. Trying to sell customers on “automated writing” in 2011 was how I imagine it felt to Henry Ford when he sold automobiles to people that had only seen horses: quizzical looks and disbelief that such a contraption was even possible—or necessary.

A short four years later and things changed dramatically. The Gartner Hype Cycle (published since 1995) has been a consistent measure of the hype surrounding emerging technologies. Take a look at the 2016 report:



Source: Gartner

Several AI-related technologies make an appearance including Smart Robots, Autonomous Vehicles, Conversational User Interfaces, Natural Language Question-Answering, and of course, Machine Learning at the *Peak of Inflated Expectations*.

We went from a general disbelief in software being able to automate things that were strictly in the domain of humans to every startup having a token AI slide in their pitch decks. Before, I had to tone down the mention of “Artificial Intelligence” so we didn’t sound too geeky and scare people away, to now customers are disappointed if your solution isn’t completely magical. Before, the quickest way to end a conversation at a dinner party was to mention AI, and now my

insurance agent tells me about this cool new “AI assistant” he’s using to schedule all his meetings.

Whether we’ve reached “Peak AI” or not, most people, including my retired mom, have heard about it. The big question remains: can it live up to the hype? How does AI compare with previous technologies regarding its *potential*? The answer is what gets me excited about the future.

The eight technology waves

I’ve identified eight broad technologies over the past 30 years that I think are useful for comparison purposes. I analyzed how quickly each technology evolved and the factors that either helped or impeded the rate of innovation. These eight are by no means the only technologies I could have chosen, but each one had a significant impact on the overall technology landscape during its peak. My intention is not to exhaustively go through every technology wave that’s ever occurred, but to provide a framework we can use to compare and contrast.

The eight I’m covering in this article include Desktop Operating Systems, Web Browsers, Networking, Social Networks, Mobile Apps, Internet of Things, Cloud Computing, and Artificial Intelligence.

Two key factors for Technology Potential

There are two factors that are very important determinants for how far and how fast a technology advances over time.

The first factor is the *barrier to entry for a single developer to create something useful*. If developers have the ability to create or tweak their own implementation, you get rapid dispersion of the technology and many improvements both big and small through the contributions of the developer crowd.

Operating Systems have a high barrier to entry because of the complexity of the software. Generally, you don’t see large numbers of developers spending time on the weekend trying to modify an operating system for some specialized purpose. On the flip side, Mobile Apps are easy to create and publish with limited skills (which has been both good and bad as I’ll explain).

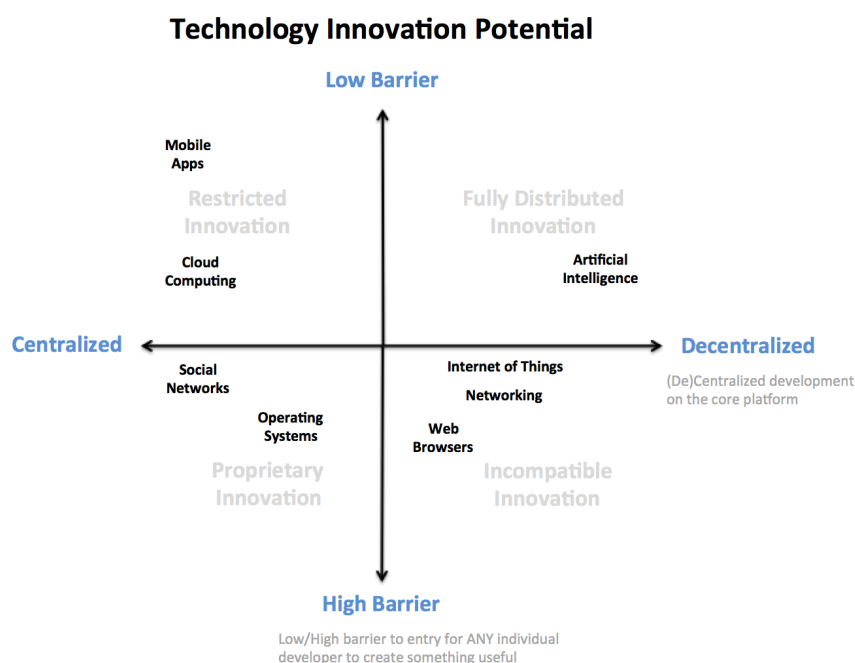
The second factor is whether *development on the core platform is centralized or decentralized*. Does one company or organization serve

as the gatekeeper for new versions of the core platform or can anyone contribute? In the Social Network space, development is completely centralized by the owners of the platforms (Twitter, Facebook, etc.) compared with Networking; the RFC process led to a widely distributed group of contributors.

A technology with a low barrier to entry and decentralized platform development has the greatest potential for future impact. None of the eight technologies possess both of those characteristics except Artificial Intelligence, but more on that later.

For the purpose of this article, I'm mainly interested in the *potential* for innovation, not how much innovation has or will occur. The two factors I mentioned aren't required for a technology to have a huge impact. Cloud Computing has had a tremendous impact even though there are only a small handful of players in the space. But the future of Cloud Computing is almost entirely dependent on the companies that own the platforms. If the market consolidates or becomes less attractive over time for the major players, innovation will cease or diminish greatly, which was the story of the Web Browser in the early 2000's.

Below is how I'd rank the eight technologies across both factors. Just like Gartner's Hype curve, the placement of each in the four quadrants was largely a subjective exercise based on my analysis. I'll go into some detail on each technology, but first I'll cover what each quadrant represents.



Low Barrier + Centralized Dev = *Restricted Innovation*

This scenario is pretty unusual because developers can easily create, but platform development is centralized. Innovation is restricted to the whims of the platform owner. Developers have to work within the confines of the platform with limited ability to influence overall direction because it is owned by a single authority. The app store model falls into this quadrant.

High Barrier + Centralized Dev = *Proprietary Innovation*

In this case, development can be done only by the platform owner. Developers have a high barrier to create and can't contribute to the platform. This is the classic proprietary model, which has been dominant in product development.

High Barrier + Decentralized Dev = *Incompatible Innovation*

This quadrant often represents technologies that are based on open source or have open (and in many cases optional) standards, but either due to being based on hardware or very complex software, the barrier for developers to create is non-trivial. The result can be significant innovation, but with solutions that are not compatible as developers are free to incorporate the "standards" they choose.

Low Barrier + Decentralized Dev = *Fully Distributed Innovation*

This is optimal in terms of an environment that foster innovation. Not only can individual developers create because the barrier to entry is low, but they can also build their own platforms. AI is the only technology I'd put in this quadrant, and I'll discuss the reasons for that in the next section.

Fully distributed innovation is difficult to scale over time because without any central coordination, you often end up with a fragmented ecosystem with many incompatible solutions. You can see this starting to occur within AI as certain frameworks become popular (e.g. Tensorflow and PyTorch).

Overview of the eight technologies

Next, I'm going to briefly describe the reasons I put each technology in a particular quadrant.

Desktop Operating Systems

The desktop operating systems have been dominated by a small number of players. IBM started in 1981 with their initial PC, Microsoft (and Sun to a lesser extent) in the 1990s, and Apple in the 2000s.

Linux was in the mix too, but could never quite go mainstream on the desktop/laptop.

Operating systems are massive pieces of software, so there isn't much that a single developer will do to improve and redistribute the core platform on their own. The complexity and breadth of code is a big reason for the high barrier to entry.

With the exception of Linux, the other popular desktop operating systems were proprietary. That means, for the most part, innovation in the OS has been centralized within the walls of big companies. Even in the case of Linux, having a benevolent dictator meant all big decisions were centrally managed. Big, monolithic software such as operating systems require centralized coordination at some level to ensure the end product is a fully integrated and stable solution.

Networking

Cisco's IPO in 1990 was the beginning of an incredible run for networking companies over the next decade. A variety of companies were created and then were acquired or merged. Cisco, Lucent, and Nortel were the big players (and eventually Juniper) before the dot-com bust took the legs out of the industry from which it never fully recovered.

Internet protocols have had a decentralized development process with Request For Comments (RFCs) since the first RFC was published in 1969. There are standards bodies, such as the IETF, which can ratify RFCs as official standards. While defining protocols was decentralized, the core platforms that made use of these protocols (e.g. Cisco routers) were still proprietary and closed.

And since the major networking vendors had their own hardware, the barrier was high for an individual developer to contribute. While it was possible for an outsider to contribute to a protocol specification, only developers at the networking companies could add those protocols to their platforms.

I include Networking in this list not so much because it was a major technology wave, but because of their unique standardization process.

Web Browser

While the operating system wars were raging on, the most important application on the desktop also went through a similar battle.

Whether it was Netscape and IE in the 1990s or Chrome, IE and

Firefox today, the browser has always been a coveted application due to it being the front-end to the web.

While IE is the only major browser that is not open source or based on open source, the barrier to entry is still high. Like operating systems, modern web browsers have become very complex pieces of software. It's not common for an average developer to fork a browser to customize it for their company's use. Each browser provides varying support for extensions or plug-ins, but these tend to be pretty limited in terms of capabilities and are often brittle.

Given the open-source nature of some browsers, development on the core platform is decentralized, but in practice, changes are closely monitored. Given the wide user base for browsers, the browser owners have to scrutinize every change or risk sending out a flawed product to millions of users.

Mobile Apps

I could list "Mobile" or "Smartphones" here instead of Mobile Apps, but many of the same issues I described under Desktop Operating Systems would apply. Mobile Apps are a little more interesting for our purposes because of their low barrier to entry. When Apple came out with the App Store, it changed things forever in Mobile computing. Similar to web pages, but more feature rich, Mobile Apps ushered in a new era of consumer capabilities.

As far as a developer creating something useful, it doesn't get much easier than a Mobile App. Some may argue that the barrier to entry is too low, so now the app stores are filled with junk. If it were a little harder to create, maybe the script kiddies and copycats of the world wouldn't have contributed, but that's the tradeoff with being open to all comers. A small percentage will create great apps, and the vast majority will not.

Social Networks

Whether it's Facebook and Twitter or Foursquare and LinkedIn, the Social Networks have all been proprietary with limited standardization. There have been some open source attempts at Social Networks, but none have gained critical mass. That means virtually all the development in the Social Network space has come from a small number of companies.

Regarding a developer's ability to create something useful, you can look at Social Networks in one of two ways, which is why I gave it a

middle of the road score on barrier to entry. It is true that unless you are an employee of one of the Social Network companies, you won't be able to contribute to their platform. On the other hand, Social networks are not nearly as complex as an operating system or web browser. The complexity comes in trying to scale Social Networks to millions (or billions) of users. A developer could cobble together a web app that resembled Facebook or Twitter in a weekend. Making them work for millions of users is a different story.

The other issue that makes Social Networks have a high barrier to entry is the required network effects before they become useful. If you don't get a critical mass of users, the network has limited value. More than just making a mobile app, obtaining network effects is extremely difficult and not something a developer can easily build on her own.

Cloud Computing

Amazon, with Google and Microsoft following fast, has done an amazing job in ushering in a new age of Cloud Computing features and pricing with AWS. However, due to the significant hardware requirements, Cloud Computing is a very capital intensive business to be successful at scale. That generally leads to centralized development of the core platform where individual companies hold the keys to the kingdom, which is the case here (Amazon, Google, Microsoft, etc.)

Cloud Computing has been optimized to allow developers to create interesting things, so the barrier to create is low. Cloud Computing has enabled a lot of innovation, but that will only continue as long as Amazon, Google, and Microsoft deem the space worth investing in.

Internet of Things

It has been interesting to watch the Internet of Things (IoT) unfold over the last ten years as it gained and lost momentum a few times. Regarding my two factors, IoT is a mixed bag. As far as barrier to entry, most of the software (and even hardware) building blocks to build an IoT device are commonly available, but taking a commercial IoT device to market is a significant undertaking—just ask me about any of the failed Kickstarter projects in which I've invested.

IoT has benefited from some standardization, but it's also a very fragmented space. It resembles the Networking environment I described earlier. Just because there are "standards" doesn't mean companies have to use them. As a result, you likely have a variety of

IoT devices in your home or office that use different standards or don't work together.

Artificial Intelligence

I first realized something was different with the AI ecosystem when I started digging into the latest research on a variety of topics: LSTMs, GANs, CNNs, Seq-to-seq. There is so much new research coming out; it's hard to keep up. Getting up-to-speed with the current state of the art and staying abreast of the latest research has been challenging, to say the least.

Looking at the two factors, first, the barrier to entry for a developer to do something interesting with AI is low. All the tools you need are freely available. The barriers are purely self-driven. To do something interesting you need a significant dataset (although there is some debate about how much you need) and the other is you need the mental bandwidth to understand how to build a useful model. It's this last point that leaves me a little skeptical about the vast number of companies that claim they are doing interesting things with AI, specifically Machine Learning. Building an ML program isn't like building a Mobile App. It's more complicated, especially if you are doing anything interesting (although Amazon has simplified the process considerably).

Regarding the core platforms, it is still early days, and there aren't heavyweight platforms that define the experience with AI like the other technologies I've covered. All of the machine learning and deep learning platforms like Tensorflow, PyTorch, Theano, and Keras are open-source and have vibrant communities.

The key that makes AI different than most other technologies is its strong research background. By default, the computer science field has been an open community marked by academic conferences where researchers present their latest findings. Many of the luminaries in the AI world hail from academia where publishing their research open was the norm. Instead of keeping innovation closed or waiting until the final moment once an idea has been fully baked and lots of code written, most research is based on a few months worth of work and limited code. The point is to get the ideas out to the community as quickly as possible so others can improve on them (and you get credit for the idea before someone else does).

The closest thing we've seen to this kind of ecosystem is the Internet standards bodies or open source movements I mentioned earlier, but

AI is still different in important ways. When it comes to AI research, there is no governing body that approves new advances. The IETF and other organizations became notorious for playing favorites to the incumbents as well as being heavily political with the big powers in the industry forcing their representation so they can steer standards groups in particular ways. The closest thing to that in the research community is the paper submission process at the big academic conferences and awards for best paper. While those accolades are nice, they are not a necessary condition for your new idea to be picked up by the research crowd.

Even with Google, Facebook, Amazon, and Baidu scooping up Ph.D. students as quickly as they can, retraining their workforce on Machine Learning, and investing billions of dollars in the space, this is one of the few times in the past 30 years where a lot of that investment is helping push the whole industry forward instead of just a particular company's agenda.

Confluence of circumstances

I often tell entrepreneurs that most startups fail too soon. They have to hang around long enough for the confluence of circumstances to line up in their favor. That's what happened in my case. There was a domino effect of companies getting acquired that led to mine getting acquired for a great return. That said, this is the most troubling aspect of startup life for me—you can't control everything that will make you successful. Market forces, technology shifts, economic conditions, etc. all have a significant impact on your company.

The same thing applies to technologies. Despite failed attempts in the past, the current AI boom came along at just the right time. There are a variety of circumstances that have contributed to its success including:

- The **fully distributed innovation** environment I've discussed has resulted in a very rapid rate of new capabilities throughout the AI field.
- After years of hype around Big Data starting in the mid-2000's, by 2010 and 2011 many companies had finally started to develop **Big Data infrastructures**.
- After focusing years on the "what" instead of the "why," **companies wanted to realize value in their Big Data investments**. The problem with Big Data is it's not an end, but a means to an end. Pent-up demand to get something meaningful

out of Big Data meant companies were open to new ways of making use of their data (enter AI stage left).

- Thanks in large part to the gaming world, **optimized compute for AI (in the form of GPUs) have become readily available.** GPUs perform matrix multiplication much faster than traditional CPUs, which means machine learning and deep learning models can perform much faster. Nvidia is happy.
- There is a **low barrier to entry to develop AI solutions, but it requires a high level of technical knowledge.** This is an important difference from Mobile Apps, which also have a low barrier to develop. Where App Stores are marred by numerous crappy apps, unless you are a serious developer with access to significant data, you won't be able to do much with Machine Learning. As a result, the average AI app will be higher quality.

Will AI be LeBron James or Greg Oden?

In this article, I've described why the potential for future innovation with Artificial Intelligence is unlike anything we've seen among previous major technology waves. However, you can have all the potential in the world and not make full use of it. Just ask Tiger Woods.

The AI ecosystem will be resilient due to the fully distributed innovation model I've described, but there are a few external factors that could impede progress.

- **Too focused on Deep Learning.** A researcher from the Allen Institute for AI gave a talk at the O'Reilly AI conference last year and shared his concern that with all the attention Deep Learning has attracted, we may develop a local maximum on DL and not explore other methods that may be better suited to help achieve general AI.
- **Not enough good Big Data.** Despite the progress many companies have made at organizing their data, most still have a long way to go. At Automated Insights, we've seen the #1 impediment to successful projects is lack of enough quality data. Most companies think they have better data than they do.
- **Humans!** Easily the biggest impediment to adoption of next-generation automation and AI technologies will be ourselves. As a society, we will not fully embrace technologies that can save millions of lives like autonomous cars. The technology for Level 5

autonomous cars is within reach, but societal and political pressures will make it take considerably longer to get fully implemented.

- **Surviving the Trough of Disillusionment.** It would be interesting to ask the Gartner folks, but I'd be curious if they've seen in the last 20 years a technology get as hyped as quickly as AI. They use the same hype curve for all emerging technologies, but in reality, each one has different slope to their curve. Given just how much hype has surrounded AI, it may mean the *Trough of Disillusionment* could also be deeper and potentially harder to come out of. Will people get soured on AI when early results don't come back as amazing as we hoped?
- **AI platforms emerge that make development more centralized.** Tensorflow has gained a lot of traction in the ML community. Could it become a more heavyweight platform that becomes the default framework every ML engineer has to use? Then we risk following a more centralized innovation pattern.

Ultimately, I believe Amara's Law will hold true for AI:

We tend to overestimate the effect of a technology in the short run and underestimate the effect in the long run.

Many manual tasks I thought were untouchable just a year or two ago, I can see a path to making them more automated. It will be interesting to watch if there is a tipping point where the rate of innovation goes even quicker as we make additional breakthroughs.

. . .

If you like this post, give it a below so others will see it too. Thank you!

