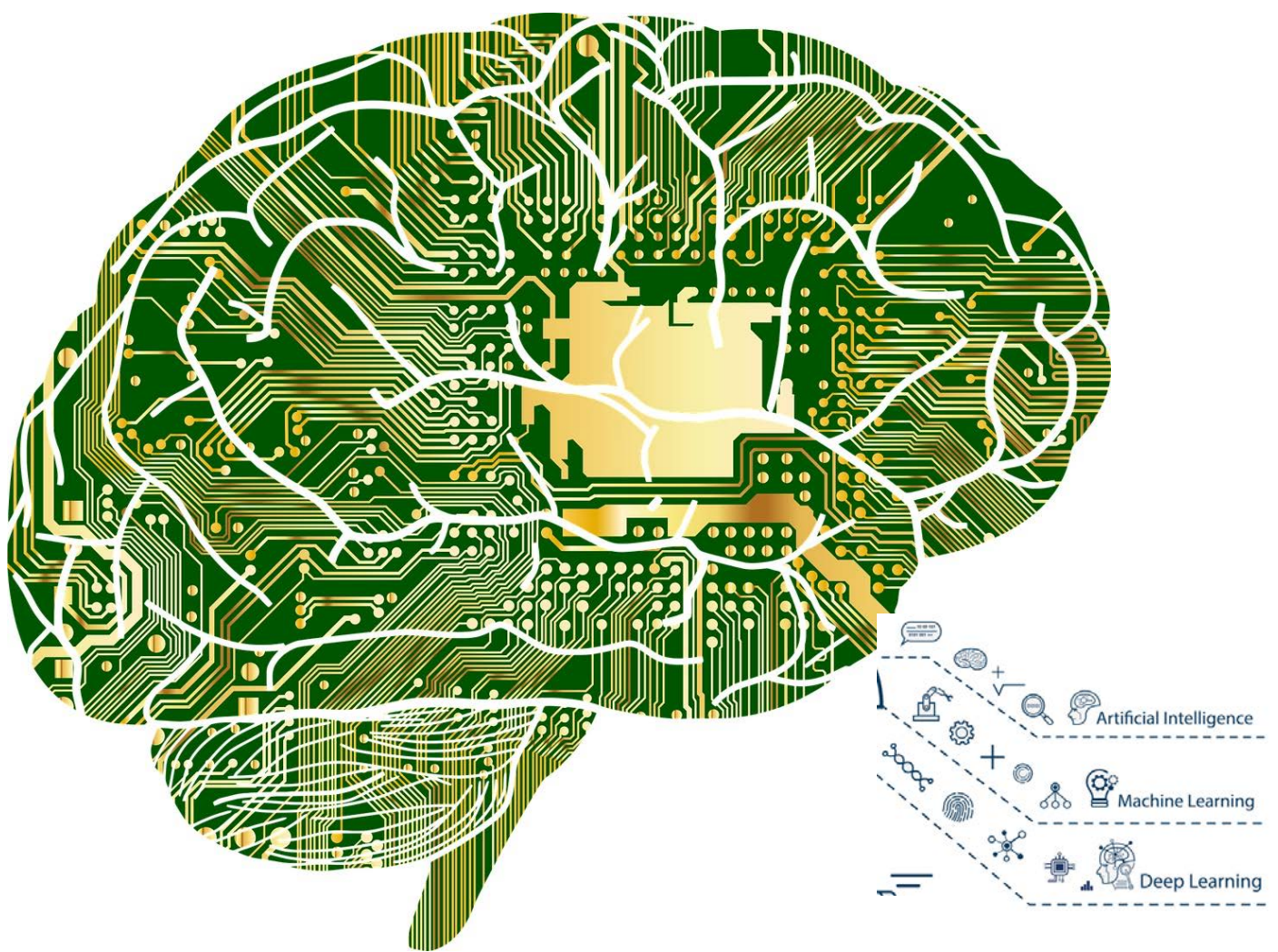


V-CARE COURSES





Artificial Intelligence/ Machine Learning Explained

Author: Steve Blank

Stanford | Gordian Knot Center for
National Security Innovation

Artificial Intelligence/Machine Learning– Explained

AI is a once-in-a-lifetime commercial and defense game changer

Hundreds of billions in public and private capital is being invested in AI and Machine Learning companies. The [number of patents filed in 2021 is more than 30 times higher than in 2015](#) as companies and countries across the world have realized that AI and Machine Learning will be a major disruptor and potentially change the balance of military power.

Until recently, the hype exceeded reality. Today, however, advances in AI in several important areas ([here](#), [here](#), [here](#), [here](#) and [here](#)) equal and even surpass human capabilities.

If you haven't paid attention, now's the time.

AI and the DoD

The Department of Defense has thought that AI is such a foundational set of technologies that they started a dedicated organization- [the JAIC](#) - to enable and implement artificial intelligence across the Department. They provide the infrastructure, tools, and technical expertise for DoD users to successfully build and deploy their AI-accelerated projects.

Some specific defense related AI applications are listed later in this document.

We're in the Middle of a Revolution

Imagine it's 1950, and you're a visitor who traveled back in time from today. Your job is to explain the impact computers will have on business, defense and society to people who are using manual calculators and slide rules. You succeed in convincing one company and a government to adopt computers and learn to code much faster than their competitors /adversaries. And they figure out how they could digitally enable their business – supply chain, customer interactions, etc. Think about the competitive edge they'd have by today in business or as a nation. They'd steamroll everyone.

That's where we are today with Artificial Intelligence and Machine Learning. These technologies will transform businesses and government agencies. Today, 100s of billions of dollars in private capital have been invested in 1,000s of AI startups. The U.S. Department of Defense has created a dedicated organization to ensure its deployment.

But What Is It?

Compared to the classic computing we've had for the last 75 years, AI has led to new types of applications, e.g. facial recognition; new types of algorithms, e.g. machine learning; new types of computer architectures, e.g. neural nets; new hardware, e.g. GPUs; new types of software developers, e.g. data scientists; all under the overarching theme of artificial intelligence. The sum of these feels like buzzword bingo. But they herald a sea change in what computers are capable of doing, how they do it, and what hardware and software is needed to do it.

This brief will attempt to describe all of it.

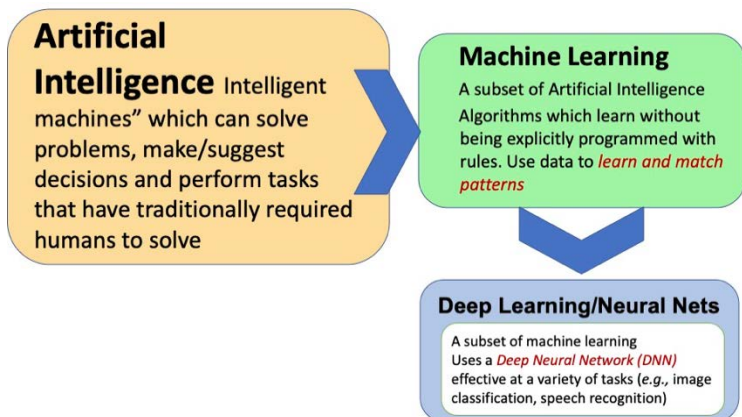
New Words to Define Old Things

One of the reasons the world of AI/ML is confusing is that it's created its own language and vocabulary. It uses new words to define programming steps, job descriptions, development tools, etc. But once you understand how the new world maps onto the classic computing world, it starts to make sense. So first a short list of some key definitions.

AI/ML - a shorthand for Artificial Intelligence/Machine Learning

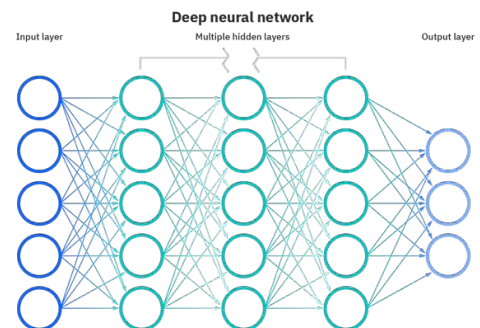
Artificial Intelligence (AI) - a catchall term used to describe "Intelligent machines" which can solve problems, make/suggest decisions and perform tasks that have traditionally required humans to do. AI is not a single thing, but a constellation of different technologies.

Machine Learning (ML) - a subfield of artificial intelligence. Humans combine data with [algorithms](#) (see [here](#) for a list) to *train a model* using that data. This trained model can then make predications on new data (is this picture a cat, a dog or a person?) or decision-making processes (like understanding text and images) without being explicitly programmed to do so.



[Machine learning algorithms](#) - computer programs that adjust themselves to perform better as they are exposed to more data. The "learning" part of machine learning means these programs change how they process data over time. In other words, a machine-learning algorithm can adjust its own settings, given feedback on its previous performance in making predictions about a collection of data (images, text, etc.).

[Deep Learning/Neural Nets](#) – a subfield of machine learning. [Neural networks](#) make up the backbone of deep learning. (The "deep" in deep learning refers to the depth of layers in a neural network.) Neural nets are effective at a variety of tasks (e.g., image classification, speech recognition). A deep learning neural net algorithm is given massive volumes of data, and a task to perform - such as classification. The resulting model is capable of solving complex tasks such as recognizing objects within an image and translating speech in real time. In reality, the neural net is a logical concept that gets mapped onto a physical set of specialized processors. See [here](#).)



Data Science – a new field of computer science. Broadly it encompasses data systems and processes aimed at maintaining data sets and deriving meaning out of them. In the context of AI, it's the practice of people who are doing machine learning.

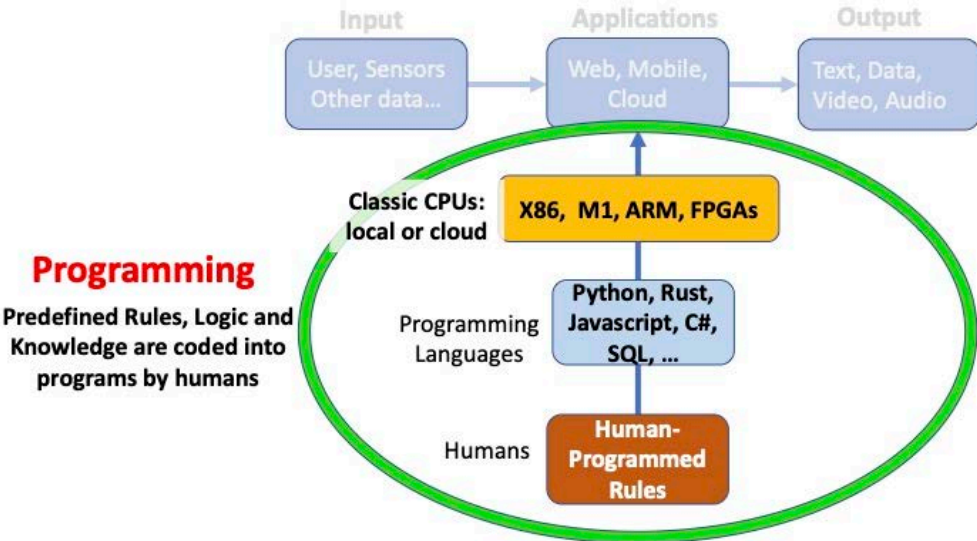
Data Scientists - responsible for extracting insights that help businesses make decisions. They explore and analyze data using machine learning platforms to create models about customers, processes, risks, or whatever they're trying to predict.

What's Different? Why is Machine Learning Possible Now?

To understand why AI/Machine Learning can do these things, let's compare them to computers before AI came on the scene. (Warning – *simplified* examples below.)

Classic Computers

For the last 75 years computers (we'll call these *classic computers*) have both shrunk to pocket size (iPhones) and grown to the size of warehouses (cloud data centers), yet they all continued to operate essentially the same way.



Classic Computers - Programming

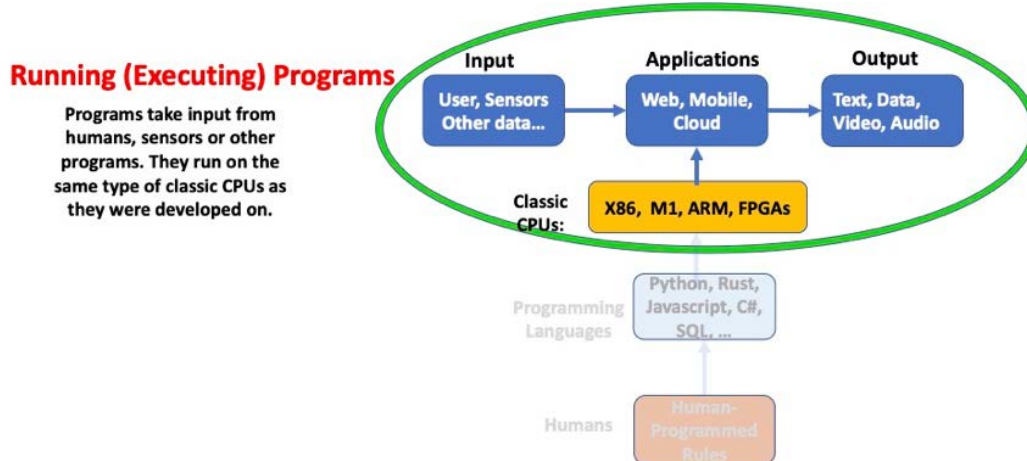
Classic computers are designed to do anything a human *explicitly tells them to do*. People (programmers) write software code (programming) to develop applications, thinking a priori about all the rules, logic and knowledge that need to be built in to an application so that it can deliver a specific result. These rules are explicitly coded into a program using a software language (Python, JavaScript, C#, Rust, ...).

Classic Computers - Compiling

The code is then *compiled* using software to translate the programmer's source code into a version that can be run on a target computer/browser/phone. For most of today's programs, the computer used to develop and compile the code does not have to be that much faster than the one that will run it.

Classic Computers - Running/Executing Programs

Once a program is coded and compiled, it can be deployed and run (executed) on a desktop computer, phone, in a browser window, a data center cluster, in special hardware, etc. Programs/applications can be games, social media, office applications, missile guidance systems, bitcoin mining, or even operating systems e.g. Linux, Windows, IOS. These programs run on the same type of classic computer architectures they were programmed in.



Classic Computers – Software Updates, New Features

For programs written for classic computers, software developers receive bug reports, monitor for security breaches, and send out regular software updates that fix bugs, increase performance and at times add new features.

Classic Computers- Hardware

The CPUs (Central Processing Units) that write and run these Classic Computer applications all have the same basic design (architecture). The CPUs are designed to handle a wide range of tasks quickly in a *serial* fashion. These CPUs range from Intel X86 chips, and the ARM cores on Apple M1 SoC, to the z15 in IBM mainframes.

Machine Learning

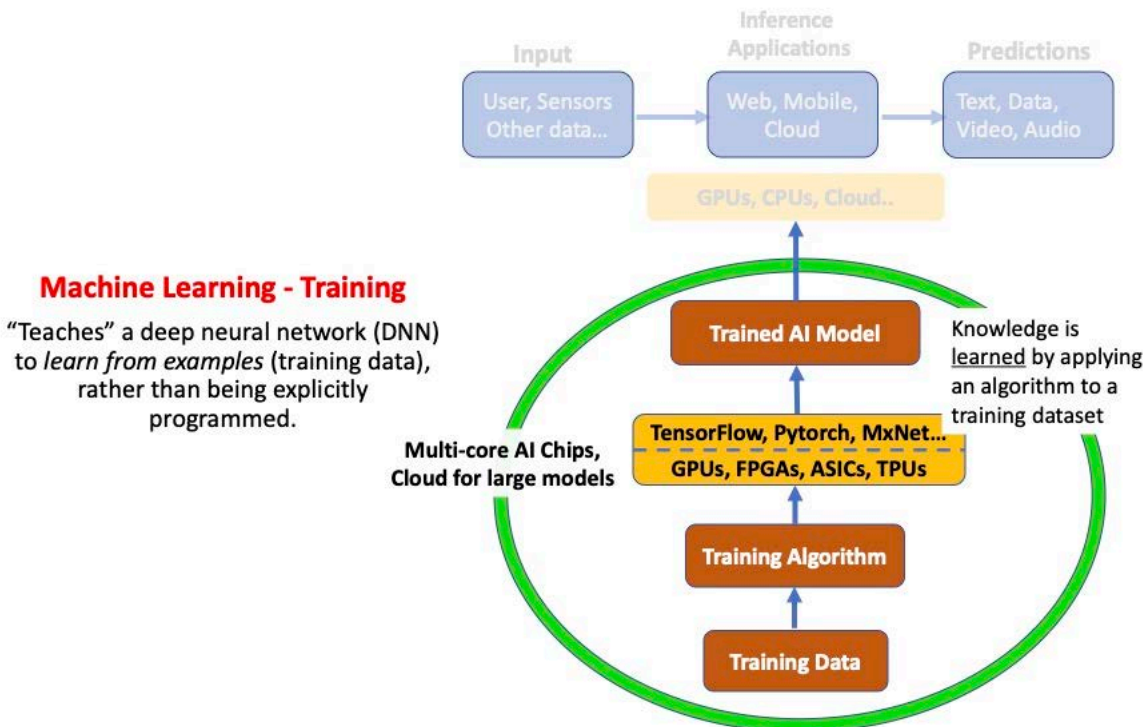
In contrast to programming on classic computing with fixed rules, machine learning is just like it sounds – we can train/teach a computer to “learn by example” by feeding it lots and lots of examples. (For images a rule of thumb is that a machine learning algorithm needs at least 5,000 labeled examples of each category in order to produce an AI model with decent performance.) Once it is trained, the computer runs on its own and can make predictions and/or complex decisions.

Just as traditional programming has three steps - first *coding* a program, next *compiling* it and then *running* it - machine learning also has three steps: *training* (teaching), *pruning* and *inference* (predicting by itself.)

Machine Learning - Training

Unlike programming classic computers with explicit rules, training is the process of “teaching” a computer to perform a task e.g. recognize faces, signals, understand text, etc. (Now you know why you're asked to click on images of traffic lights, cross walks, stop signs, and buses or type the text of scanned image in [ReCaptcha](#).) Humans provide massive volumes of “training data” (the more data, the better the model’s performance) and select the appropriate algorithm to find the best optimized outcome.

(See the detailed “machine learning pipeline” later in this section for the gory details.)



By running an algorithm selected by a data scientist on a set of training data, the Machine Learning system generates the rules embedded in a trained model. *The system learns from examples* (training data), rather than being explicitly programmed. (See the “Types of Machine Learning” section for more detail.) This self-correction is pretty cool. An input to a neural net results in a guess about what that input is. The neural net then takes its guess and compares it to a ground-truth about the data, effectively asking an expert “Did I get this right?” The difference between the network’s guess and the ground truth is its *error*. The network measures that error, and walks the error back over its model, adjusting weights to the extent that they contributed to the error.)

Just to make the point again: *The algorithms combined with the training data - not external human computer programmers - create the rules that the AI uses.* The resulting model is capable of solving complex tasks such as recognizing objects it’s never seen before, translating text or speech, or controlling a drone swarm.

(Instead of building a model from scratch you can now buy, for common machine learning tasks, *pretrained models* [from others](#) and [here](#), much like chip designers buying [IP Cores](#).)

Machine Learning Training - Hardware

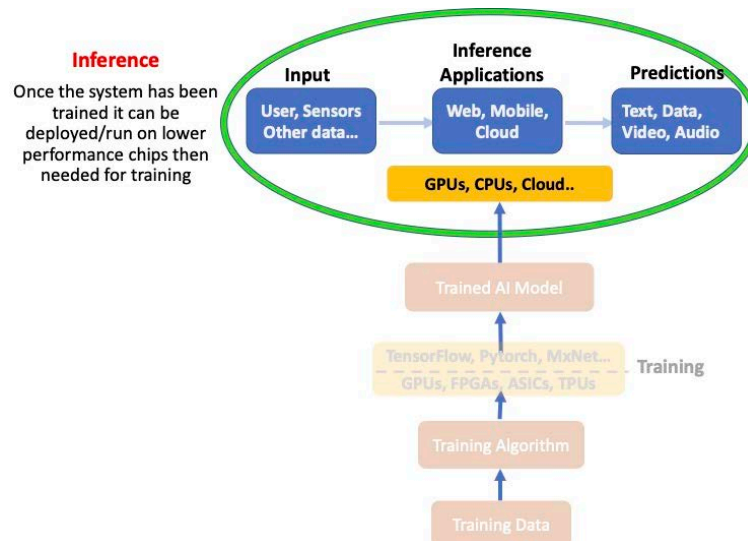
Training a machine learning model is a very computationally intensive task. AI hardware must be able to perform thousands of multiplications and additions in a mathematical process called matrix multiplication. It requires specialized chips to run fast. (See the AI hardware section for details.)

Machine Learning - Simplification via pruning, quantization, distillation

Just like classic computer code needs to be compiled and optimized before it is deployed on its target hardware, the machine learning models are simplified and modified ([pruned](#)) to use less computing power, energy, and memory before they're deployed to run on their hardware.

Machine Learning – Inference Phase

Once the system has been trained it can be copied to other devices and run. And the computing hardware can now make inferences (predictions) on new data that the model has never seen before.



Inference can even occur locally on edge devices where physical devices meet the digital world (routers, sensors, IOT devices), close to the source of where the data is generated. This reduces network bandwidth issues and eliminates latency issues.

Machine Learning Inference - Hardware

Inference (running the model) requires substantially less compute power than training. But inference also benefits from specialized AI chips.

Machine Learning – Performance Monitoring and Retraining

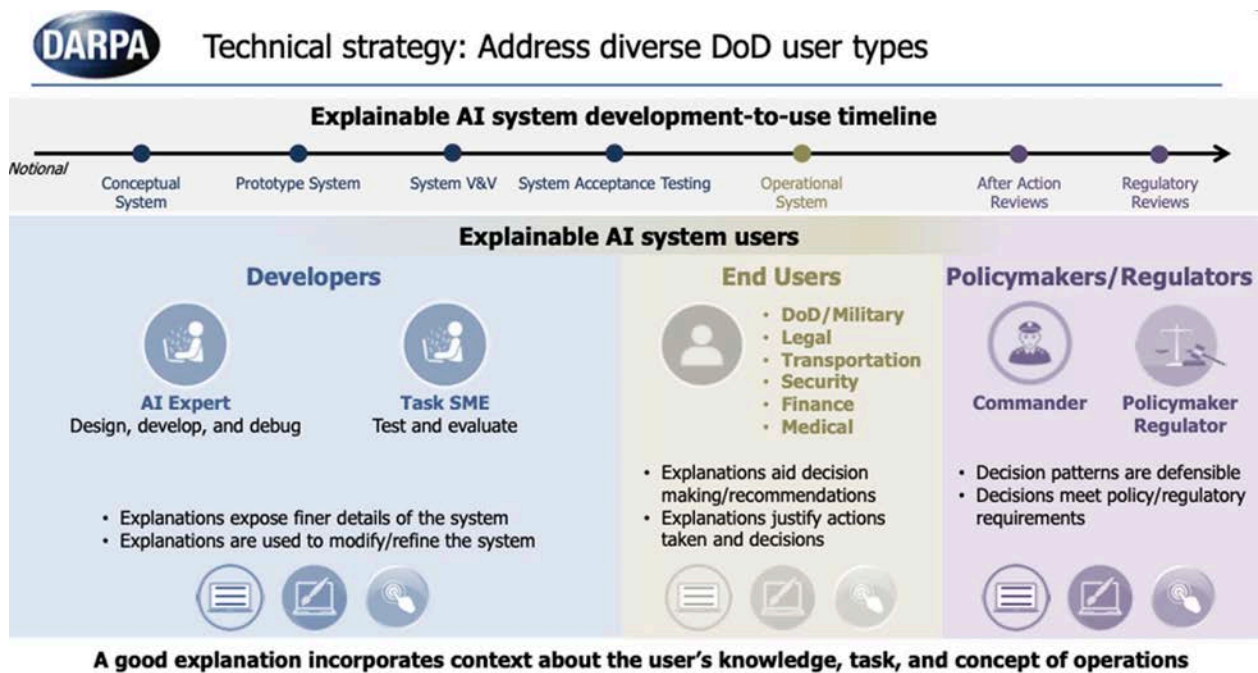
Just like classic computers where software developers do regular software updates to fix bugs and increase performance and add features, machine learning models also need to be updated regularly by adding new data to the old training pipelines and running them again. Why?

Over time machine learning models get stale. Their real-world performance generally degrades over time if they are not updated regularly with new training data that matches the changing state of the world. The models need to be monitored and retrained regularly for [data and/or concept drift](#), harmful predictions, performance drops, etc. To stay up to date, the models need to re-learn the patterns by looking at the most recent data that better reflects reality.

One Last Thing – “Verifiability/Explainability”

Understanding how an AI works is essential to fostering trust and confidence in AI production models.

Neural Networks and Deep Learning differ from other types of Machine Learning algorithms in that they have low explainability. They can generate a prediction, but it is very difficult to understand or explain how it arrived at its prediction. This “explainability problem” is often described as a problem for all of AI, but it’s primarily a problem for Neural Networks and Deep Learning. Other types of Machine Learning algorithms – for example [decision trees](#) – have very high explainability. The results of the five-year DARPA Explainable AI Program (XAI) are worth reading [here](#).



So What Can Machine Learning Do?¹

It’s taken decades but as of today, on its simplest implementations, machine learning applications can do some tasks better and/or faster than humans. Machine Learning is most advanced and widely applied today in processing text (through Natural Language Processing)

followed by understanding images and videos (through Computer Vision) and analytics and anomaly detection. For example:

Recognize and Understand Text/Natural Language Processing

AI is better than humans on basic reading comprehension benchmarks like [SuperGLUE](#) and [SQuAD](#) and their performance on complex linguistic tasks is almost there. Applications: [GPT-3](#), [M6](#), [OPT-175B](#), [Google Translate](#), Gmail Autocomplete, Chatbots, Text summarization.

Write Human-like Answers to Questions and Assist in Writing Computer Code

An AI can write original text that is indistinguishable from that created by humans. Examples [GPT-3](#), [Wu Dao 2.0](#) or generate computer code. Example [GitHub Copilot](#), [Wordtune](#)

Recognize and Understand Images and video streams



An AI can see and understand what it sees. It can identify and detect an object or a feature in an image or video. It can even identify faces. It can scan news broadcasts or read and assess text that appears in videos. It has uses in threat detection - airport security, banks, and sporting events. In medicine to [interpret MRI's](#) or to [design drugs](#). And in retail to scan and analyze in-store imagery to intuitively determine inventory movement. Examples of ImageNet benchmarks [here](#) and [here](#)

Detect Changes in Patterns/Recognize Anomalies



An AI can recognize patterns which don't match the behaviors expected for a particular system, out of millions of different inputs or transactions. These applications can discover evidence of an attack on financial networks, fraud detection in insurance filings or credit card purchases; identify fake reviews; even tag sensor data in industrial facilities that mean there's a safety issue. Examples [here](#), [here](#) and [here](#).

Power Recommendation Engines



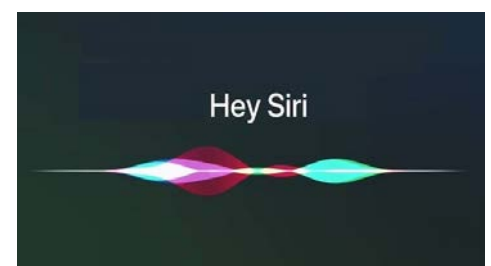
An AI can provide recommendations based on user behaviors used in ecommerce to provide accurate suggestions of products to users for future purchases based on their shopping history. Examples: [Alexa](#) and [Siri](#)

Recognize and Understand Your Voice

An AI can understand spoken language. Then it can comprehend what is being said and in what context. This can enable chatbots to have a conversation with people. It can record and transcribe meetings. (Some versions can even read lips to increase accuracy.) Examples: Siri/Alexa/Google Assistant. Example [here](#)

Create Artificial Images

AI can create artificial images ([DeepFakes](#)) that are indistinguishable from real ones using [Generative Adversarial](#)



[Networks](#). Useful in entertainment, virtual worlds, gaming, fashion design, etc. Synthetic faces are now indistinguishable and more trustworthy than photos of real people. Paper [here](#).

Create Artist Quality Illustrations from A Written Description

AI can generate images from text descriptions, creating anthropomorphized versions of animals and objects, combining unrelated concepts in plausible ways. An example is [Dall-E](#)

Generative Design of Physical Products

Engineers can input design goals into AI-driven [generative design software](#), along with parameters such as performance or spatial requirements, materials, manufacturing methods, and cost constraints. The software explores all the possible permutations of a solution, quickly generating design alternatives Example [here](#).

Sentiment Analysis

An AI leverages deep natural language processing, text analysis, and computational linguistics to gain insight into customer opinion, [understanding of consumer sentiment](#), and measuring the impact of marketing strategies. Examples: [Brand24](#), [MonkeyLearn](#)



What Does this Mean for Businesses?

Skip this section if you're interested in national security applications

Hang on to your seat. We're just at the beginning of the revolution. The next phase of AI, powered by ever increasing powerful AI hardware and cloud clusters, will combine some of these basic algorithms into applications that do things no human can. It will transform business and defense in ways that will create new applications and opportunities.

Human-Machine Teaming

Applications with embedded intelligence have already begun to appear thanks to massive language models. For example - [Copilot as a pair-programmer](#) in Microsoft Visual Studio VSCode. It's not hard to imagine [DALL-E 2](#) as an illustration assistant in a photo editing application, or [GPT-3 as a writing assistant](#) in Google Docs.

AI in Medicine

AI applications are already appearing in radiology, dermatology, and oncology. Examples: [IDx-DR](#), [OsteoDetect](#), [Embrace2](#). AI Medical image identification can automatically detect lesions, and tumors with diagnostics equal to or greater than humans. For Pharma, AI will power [drug discovery design](#) for finding new drug candidates. The FDA has a plan for approving AI software [here](#) has a list of AI-enabled medical devices [here](#).

Autonomous Vehicles

Harder than it first seemed, but car companies like [Tesla](#) will eventually get better than human autonomy for highway driving and eventually city streets.

Decision support

Advanced virtual assistants can listen to and observe behaviors, build and maintain data models, and predict and recommend actions to assist people with and automate tasks that were previously only possible for humans to accomplish.

Supply chain management

AI applications are already appearing in predictive maintenance, risk management, procurement, order fulfillment, supply chain planning and promotion management.

Marketing

AI applications are already appearing in real-time personalization, content and media optimization and campaign orchestration to augment, streamline and automate marketing processes and tasks constrained by human costs and capability, and to uncover new customer insights and accelerate deployment at scale.

Making business smarter: Customer Support

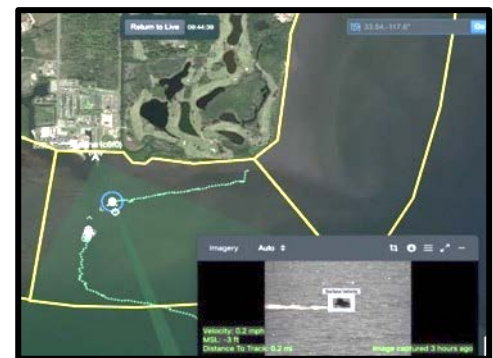
AI applications are already appearing in virtual customer assistants with speech recognition, sentiment analysis, automated/augmented quality assurance and other technologies providing customers with 24/7 self- and assisted-service options across channels.

AI in National Security²

Much like the dual-use/dual-nature of classical computers AI developed for commercial applications can also be used for national security.

AI/ML and Ubiquitous Technical Surveillance

AI/ML have made most cities [untenable for traditional tradecraft](#). Machine learning can integrate travel data (customs, airline, train, car rental, hotel, license plate readers...,) integrate feeds from CCTV cameras for facial recognition and gait recognition, breadcrumbs from wireless devices and then combine it with DNA sampling. The result is automated persistent surveillance.



China's employment of [AI as a tool of repression and surveillance](#) of the Uyghurs is a dystopian of how a totalitarian regimes will use AI-enable ubiquitous surveillance to repress and monitor its own populace.

AI/ML on the Battlefield

AI will enable new levels of performance and autonomy for weapon systems. *Autonomously collaborating assets* (e.g., drone swarms, ground vehicles) that can coordinate attacks, ISR missions, & more.

Fusing and making sense of sensor data (detecting threats in optical /SAR imagery, classifying aircraft based on radar returns, searching for anomalies in radio frequency signatures, etc.) Machine learning is better and faster than humans in finding targets hidden in a high-clutter background. Automated target detection and fires from satellite/UAV.

For example, an Unmanned Aerial Vehicle (UAV) or Unmanned Ground Vehicles with on board AI edge computers could [use deep learning to detect and locate concealed chemical, biological and explosives](#) threats by fusing imaging sensors and chemical/biological sensors.

Other examples include:

Use AI/ML countermeasures against adversarial, low probability of intercept/low probability of detection ([LPI/LPD radar techniques](#)) in radar and communication systems.

Given sequences of observations of unknown radar waveforms from arbitrary emitters without a priori knowledge, [use machine learning to develop behavioral models to enable inference of radar intent and threat level](#), and to enable prediction of future behaviors.

For objects in space, [use machine learning to predict and characterize a spacecrafts possible actions](#), its subsequent trajectory, and what threats it can pose from along that trajectory. Predict the outcomes of finite burn, continuous thrust, and impulsive maneuvers.

AI empowers other applications such as:

- [Flight Operations Planning Decision Aid Tool](#) for Strike Operations Aboard Aircraft Carriers
- [Automated Battle management](#) – air and missile defense, army/navy tactical...

AI/ML in Collection

The front end of intelligence collection platforms has created a firehose of data that have overwhelmed human analysts. “Smart” *sensors* coupled with inference engines that can pre-process raw intelligence and prioritize what data to transmit and store –helpful in degraded or low-bandwidth environments.

Human-Machine Teaming in Signals Intelligence

Applications with embedded intelligence have already begun to appear in commercial applications thanks to massive language models. For example - [Copilot as a pair-programmer](#) in

Microsoft Visual Studio VSCode. It's not hard to imagine an AI that can detect and isolate anomalies and other patterns of interest in all sorts of signal data faster and more reliably than human operators.

AI-enabled natural language processing, computer vision, and audiovisual analysis can vastly reduce manual data processing. Advances in speech-to-text transcription and language analytics now enable reading comprehension, question answering, and automated summarization of large quantities of text. This not only prioritizes the work of human analysts, it's a major force multiplier



AI can also be used to automate data conversion such as translations and decryptions, accelerating the ability to derive actionable insights.

Human-Machine Teaming in *Tasking and Dissemination*

AI-enabled systems will automate and optimize tasking and collection for platforms, sensors, and assets in near-real time in response to dynamic intelligence requirements or changes in the environment.

AI will be able to automatically generate machine-readable versions of intelligence products and disseminate them at machine speed so that computer systems across the IC and the military can ingest and use them in real time without manual intervention.

Human-Machine Teaming in *Exploitation and Analytics*

AI-enabled tools can augment filtering, flagging, and triage across multiple data sets. They can identify connections and correlations more efficiently and at a greater scale than human analysts, and can flag those findings and the most important content for human analysis. AI can fuse data from multiple sources, types of intelligence, and classification levels to produce accurate predictive analysis in a way that is not currently possible. This can improve indications and warnings for military operations and active cyber defense.

AI/ML *Information warfare*

Nation states have used AI systems to enhance disinformation campaigns and cyberattacks. This included using "[DeepFakes](#)" (fake videos generated by a neural network that are nearly indistinguishable from reality). They are harvesting data on Americans to build profiles of our beliefs, behavior, and biological makeup for tailored attempts to manipulate or coerce individuals.

But because a large percentage of it is open-source AI is not limited to nation states, AI-powered cyber-attacks, deepfakes and AI software paired with commercially available drones can create "poor-man's smart weapons" for use by rogue states, terrorists and criminals.

AI/ML Cyberwarfare

AI-enabled malware can learn and adapt to a system's defensive measures, or, conversely, AI-enabled cyber-defensive tools can proactively locate and address network anomalies and system vulnerabilities.

| Current Threats Advanced BY AI Systems | New Threats FROM AI Systems | Threats TO AI Stacks Themselves | Future Threats VIA AI Systems |
|--|---|--|---|
| <p>AI transforms existing range and reach of threats</p> <ul style="list-style-type: none">• Self-replicating AI-generated malware• Improved and autonomous disinformation campaigns• AI-engineered and targeted pathogens | <p>AI creates new threat phenomena</p> <ul style="list-style-type: none">• Deepfakes and computational propaganda• Micro-targeting: AI-fused data for targeting or blackmail• AI swarms and nano-swarms | <p>AI itself is also a new attack surface</p> <ul style="list-style-type: none">• AI attack involves the whole "AI stack". Examples include:<ul style="list-style-type: none">◦ Model inversion◦ Training data manipulation◦ "Data lake" poisoning | <p>Examples of potential threats to keep in view</p> <ul style="list-style-type: none">• Rapid machine-to-machine escalation via automated C2• AI-enabled human augmentation by peer competitors• Proliferation of simple lethal autonomous weapons to terrorists |

Source: Final Report: National Security Commission on Artificial Intelligence

AI-driven malware, where a malicious logic embeds machine learning methods and models to automatically: (i) probe the target system for inferring actionable intelligence (e.g. system configuration or operational patterns) and (ii) customize the attack payload accordingly (e.g. determine the most opportune time to execute the payload so to maximize the impact).



Attacks Against AI - *Adversarial AI*

As AI proliferates, defeating adversaries will be predicated on [defeating their AI and vice versa](#). As Neural Networks take over sensor processing and triage tasks, a human may only be alerted if the AI deems it suspicious. Therefore, we only need to defeat the AI to evade detection, not necessarily a human.

Adversarial attacks against AI fall into three types:

- [Data misclassification](#)- to generate false positive or negative results
- Synthetic data generation-to feed false information
- Data analysis – for AI-assisted classical attack generation

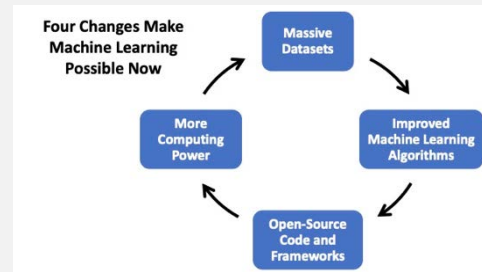
AI Attack Surfaces

Electronic Attack (EA), Electronic Protection (EP), Electronic Support (ES) all have analogues in the AI algorithmic domain. In the future, we may play the same game about the "Algorithmic Spectrum," denying our adversaries their AI capabilities while defending ours. Other can steal or poison our models or manipulate our training data.

What Makes AI Possible Now?³

Four changes make Machine Learning possible now:

1. Massive Data Sets
2. Improved Machine Learning algorithms
3. Open-Source Code, Pretrained Models and Frameworks
4. More computing power



Massive Data Sets

Machine Learning algorithms tend to require large quantities of training data in order to produce high-performance AI models. (Training Google's GPT-3 Natural Language Model with 175 billion parameters takes 1,024 [Nvidia A100 GPUs](#) more than one month.) Today, strategic and tactical sensors pour in a firehose of images, signals and other data. Billions of computers, digital devices and sensors connected to the Internet, producing and storing large volumes of data, which provide other sources of intelligence. For example facial recognition requires millions of labeled images of faces for training data.

Of course more data only helps if the data is relevant to your desired application. Training data needs to match the real-world operational data very, very closely to train a high-performing AI model.

Improved Machine Learning algorithms

The first Machine Learning algorithms are decades old, and some remain incredibly useful. However, researchers have discovered new algorithms that have greatly sped up the fields cutting-edge. These new algorithms have made Machine Learning models more flexible, more robust, and more capable of solving different types of problems.

Open-Source Code, Pretrained Models and Frameworks

Developing Machine Learning systems required a lot of expertise and custom software development that made it out of reach for most organizations. Now open-source code libraries and developer tools allow organizations to use and build upon the work of external communities. No team or organization has to start from scratch, and many parts that used to require highly specialized expertise have been automated. Even non-experts and beginners can create useful AI tools. In some cases, open-source ML models can be entirely reused and purchased. Combined with standard competitions, open source, pretrained models and frameworks have moved the field forward faster than any federal lab or contractor. It's been a feeding frenzy with the best and brightest researchers trying to one-up each other to prove which ideas are best.

The downside is that, unlike past DoD technology development - where the DoD leads it, can control it, and has the most advanced technology (like stealth and electronic warfare), in most cases the DoD will not have the most advanced algorithms or models. The analogy for AI is closer to microelectronics than it is EW. The path forward for the DoD should be supporting open research, but optimizing on data set collection, harvesting research results, and fast application.

More computing power – special chips

Machine Learning systems require a lot of computing power. Today, it's possible to run Machine Learning algorithms on massive datasets using commodity Graphics Processing Units (GPUs). (See the machine learning hardware section below). While many of the AI performance improvements have been due to human cleverness on better models and algorithms, most of the performance gains have been the massive increase in compute performance. (See the semiconductor section.)

More computing power – AI In the Cloud

The rapid growth in the size of machine learning models has been achieved by the move to large data center clusters. The size of machine learning models are limited by time to train them. For example, in training images, the size of the model scales with the number of pixels in an image. ImageNet Model sizes are 224x224 pixels. But HD (1920x1080) images require 40x more computation/memory. Large Natural Language Processing models - e.g. summarizing articles, English-to-Chinese translation like Google's GPT-3 require enormous models. GPT-3 uses 175 billion parameters and was trained on a cluster with 1,024 [Nvidia A100 GPUs](#) that cost ~\$25 million! (Which is why large clusters exist in the cloud, or the largest companies/government agencies.) [Facebooks Deep Learning and Recommendation Model \(DLRM\)](#) was trained on 1TB data and has 24 billion parameters. Some cloud vendors train on >10TB data sets.

Instead of investing in massive amounts of computers needed for training companies can use the enormous on-demand, off-premises hardware in the cloud (e.g. Amazon AWS, Microsoft Azure) for both training machine learning models and deploying inferences.

We're Just Getting Started

The next 10 years will see a massive improvement on AI inference and training capabilities. This will require regular refreshes of the hardware— on the chip and cloud clusters - to take advantage. This is the AI version of [Moore's Law](#) on steroids – applications that are completely infeasible today will be easy in 5 years.

What Can't AI Do?

While AI can do a lot of things better than humans when focused on a narrow objective, there are many things it still can't do. AI works well in specific domain where you have lots of data, time/resources to train, domain expertise to set the right goals/rewards during training, but that is not always the case.

For example AI models are only as good as the fidelity and quality of the training data. Having bad labels can wreak havoc on your training results. Protecting the integrity of the training data is critical.

In addition, AI is easily fooled by out-of-domain data (things it hasn't seen before). This can happen by "overfitting" - when a model trains for too long on sample data or when the model is too complex, it can start to learn the "noise," or irrelevant information, within the dataset.⁴ When the model memorizes the noise and fits too closely to the training set, the model becomes "[overfitted](#)," and it is unable to generalize well to new data. If a model cannot generalize well to new data, then it will not be able to perform the classification or prediction tasks it was intended for. However, if you pause too early or exclude too many important features, you may encounter the opposite problem, and instead, you may "underfit" your model. Underfitting occurs when the model has not trained for enough time, or the input variables are not significant enough to determine a meaningful relationship between the input and output variables.

AI is also poor at estimating uncertainty /confidence (and explaining its decision-making). It can't choose its own goals. (Executives need to define the decision that the AI will execute. Without well-defined decisions to be made, data scientists will waste time, energy and money.) Except for simple cases an AI can't (yet) figure out cause and effect or why something happened. It can't think creatively or apply common sense.

AI is not very good at creating a strategy (unless it can pull from previous examples and mimic them, but then fails with the unexpected.) And it lacks generalized intelligence e.g. that can generalize knowledge and translate learning across domains.

All of these are research topics actively being worked on. Solving these will take a combination of high-performance computing, advanced AI/ML semiconductors, creative machine learning implementations and [decision science](#). Some may be solved in the next decade, at least to a level where a human can't tell the difference.

Where is AI in Business Going Next?

Skip this section if you're interested in national security applications

Just as classic computers were applied to a broad set of business, science and military applications, AI is doing the same. AI is exploding not only in research and infrastructure (which go wide) but also in the application of AI to vertical problems (which go deep and depend more than ever on expertise). Some of the new applications on the horizon include Human AI/Teaming (AI helping in programming and decision making), smarter robotics and autonomous vehicles, AI-driven drug discovery and design, healthcare diagnostics, chip [electronic design automation](#), and basic science research.

Advances in language understanding are being pursued to create systems that can summarize complex inputs and engage through human-like conversation, a critical component of next-generation teaming.

Where is AI and National Security Going Next?

In the near future AI may be able to predict the future actions an adversary could take and the actions a friendly force could take to counter these. The 20th century model loop of Observe–Orient–Decide and Act (OODA) is retrospective; an observation cannot be made until after the event has occurred. An AI-enabled decision-making cycle might be ‘sense–predict–agree–act’: AI senses the environment; predicts what the adversary might do and offers what a future friendly force response should be; the human part of the human–machine team agrees with this assessment; and AI acts by sending machine-to-machine instructions to the small, agile and many autonomous warfighting assets deployed en masse across the battlefield.

An example of this is [DARPA’s ACE \(Air Combat Evolution\) program](#) that is developing a warfighting concept for combined arms using a manned and unmanned systems. Humans will [fight in close collaboration with autonomous weapon systems](#) in complex environments with tactics informed by artificial intelligence.

A Once-in-a-Generation Event

Imagine it’s the 1980’s and you’re in charge of an intelligence agency. SIGINT and COMINT were analog and RF. You had worldwide collection systems with bespoke systems in space, air, underwater, etc. And you wake up to a world that shifts from copper to fiber. Most of your people, and equipment and equipment are going to be obsolete, and you need to learn how to capture those new bits. Almost every business processes needed to change, new organizations needed to be created, new skills were needed, and old ones were obsoleted. *That’s what AI/ML is going to do to you and your agency.*

The primary obstacle to innovation in national security is not technology, it is culture. The DoD and IC must overcome a host of institutional, bureaucratic, and policy challenges to adopting and integrating these new technologies. Many parts of our culture are resistant to change, reliant on traditional tradecraft and means of collection, and averse to risk-taking, (particularly acquiring and adopting new technologies and integrating outside information sources.)

History tells us that late adopters fall by the wayside as more agile and opportunistic governments master new technologies.

Carpe Diem.

Want more Detail?

Read on if you want to know about Machine Learning chips, see a sample Machine Learning Pipeline and learning about the four types of Machine Learning.

Artificial Intelligence/Machine Learning *Semiconductors*

Skip this section if all you need to know is that special chips are used for AI/ML.

AI/ML, semiconductors, and high-performance computing are intimately intertwined - and progress in each is dependent on the others. (See the "[Semiconductor Ecosystem](#)" report.)

Some machine learning models can have [trillions of parameters](#) and require a massive number of specialized AI chips to run. Edge computers are significantly less powerful than the massive compute power that's located at data centers and the cloud. They need low power and specialized silicon.

Why Dedicated AI Chips and Chip Speed Matter

Dedicated chips for neural nets (e.g. [Nvidia GPUs](#), Xilinx FPGAs, Google TPUs) are faster than conventional CPUs for three reasons: 1) they use parallelization, 2) they have larger memory bandwidth and 3) they have fast memory access.

There are three types of AI Chips:

- Graphics Processing Units (GPUs) - Thousands of cores, parallel workloads, widespread use in machine learning
- Field-Programmable Gate Arrays (FPGAs) - Good for algorithms; compression, video encoding, cryptocurrency, genomics, search. Needs specialists to program,
- Application-Specific Integrated Circuits (ASICs) – custom chips e.g. Google TPU's



[Matrix multiplication](#) plays a big part in neural network computations, especially if there are many layers and nodes. Graphics Processing Units (GPUs) contain 100s or 1,000s of cores that can do these multiplications simultaneously. And neural networks are inherently parallel which means that it's easy to run a program across the cores and clusters of these processors. That makes AI chips 10s or even 1,000s of times faster and more efficient than classic CPUs for training and inference of AI algorithms. State-of-the-art AI chips are dramatically more cost-effective than state-of-the-art CPUs as a result of their greater efficiency for AI algorithms.

Cutting-edge AI systems require not only AI-specific chips, but state-of-the-art AI chips. Older AI chips incur huge energy consumption costs that quickly balloon to unaffordable levels. Using older AI chips today means overall costs and slowdowns at least an order of magnitude greater than for state-of-the-art AI chips.

Cost and speed make it virtually impossible to develop and deploy cutting-edge AI algorithms without state-of-the-art AI chips. Even with state-of-the-art AI chips, training a large AI algorithm can cost tens of millions of dollars and take weeks to complete. With general-purpose chips like CPUs or older AI chips, this training would take much longer and cost orders of magnitude more, making staying at the R&D frontier impossible. Similarly, performing inference using less advanced or less specialized chips could involve similar cost overruns and take orders of magnitude longer.



In addition to off-the-shelf AI chips from Nvidia, Xilinx and Intel, large companies like Facebook, Google, Amazon, have designed their own chips to accelerate AI. The opportunity is so large that there are *hundreds* of AI accelerator startups designing their own chips, funded by 10's of billions of venture capital and private equity. None of these companies own a chip manufacturing plant (a fab) so they all use a foundry (an independent company that makes chips for others) like TSMC in Taiwan (or SMIC in China for Defense related silicon.)

A Sample of AI GPU, FPGA and ASIC AI Chips and Where They're Made

| Type | Country | Company | AI Chip | Fab | Node |
|------|---------|------------------|--------------------|-------|------|
| GPU | | AMD | MI200 | TSMC | 6 |
| | | Nvidia | Ampere | TSMC | 7 |
| FPGA | | Jingjia | JM9271 | ? | 28 |
| | | Intel | Agilex | Intel | 10 |
| | | Xilinx | Versal/Vitis | TSMC | 16/7 |
| | | Efinix | Trion | SMIC | 40 |
| | | Gowin | Littlebee | TSMC | 55 |
| | | Shenzhen Pango | Titan | ? | 40 |
| ASIC | | Cerebras | Wafer Scale Engine | TSMC | 7 |
| | | Google | TPU v4 | TSMC | 7 |
| | | Intel | Habana | TSMC | 16 |
| | | Huawei | MLU100 | TSMC | 7 |
| | | Horizon Robotics | Journey 2 | TSMC | 28 |
| | | Intellifusion | NNP200 | ? | 22 |

IP (Intellectual Property) Vendors Also Offer AI Accelerators

AI chip designers can buy AI [IP Cores](#) – prebuilt AI accelerators from Synopsys ([EV7x](#)), Cadence ([Tensilica AI](#)), Arm ([Ethos](#)), Ceva ([SensPro2](#), [NeuPro](#)), Imagination ([Series4](#)), ThinkSilicon ([Neox](#)), FlexLogic ([eFPGA](#)), [Edgecortix](#) and others.

Other AI Hardware Architectures

[Spiking Neural Networks \(SNN\)](#) is a completely different approach from Deep Neural Nets. A form of [Neuromorphic computing](#) it tries to emulate how a brain works. SNN neurons use

simple counters and adder—no matrix multiply hardware is needed and power consumption is much lower. SNNs are good at unsupervised learning – e.g. detecting patterns in unlabeled data streams. Combined with their low power they’re a good fit for sensors at the edge. Examples: [BrainChip](#), [GrAI Matter](#), [Innatera](#), [Intel](#).

Analog Machine Learning AI chips use analog circuits to do the matrix multiplication in memory. The result is extremely low power AI for always-on sensors. Examples: Mythic ([AMP](#)), Aspinity ([AML100](#)), [Tetramem](#).

Optical (Photonics) AI Computation promise performance gains over standard digital silicon, and some are nearing production. They use intersecting coherent light beams rather than switching transistors to perform matrix multiplies. Computation happens in picoseconds and requires only power for the laser. (Though off-chip digital transitions still limit power savings.) Examples: [Lightmatter](#), [Lightelligence](#), [Luminous](#), [Lighton](#).

AI Hardware for the Edge

As more AI moves to the edge, the Edge AI accelerator market is segmenting into high-end chips for camera-based systems and low-power chips for simple sensors. For example:

AI Chips in Autonomous vehicles, Augmented Reality and multicamera surveillance systems
These inference engines require high performance. Examples: Nvidia ([Orin](#)), AMD ([Versal](#)), Qualcomm ([Cloud AI 100](#)), and acquired [Arriver](#) for automotive software.

AI Chips in Cameras for facial recognition, surveillance. These inference chips require a balance of processing power with low power. Putting an AI chip in each camera reduces latency and bandwidth. Examples: [Hailo-8](#), [Ambarella CV5S](#), Quadric ([Q16](#)), (RealTek [3916N](#)).

Ultralow-Power AI Chips Target IoT Sensors - IoT devices require very simple neural networks and can run for years on a single battery. Example applications: Presence detection, wakeword detection, gunshot detection... Examples: Syntiant ([NDP](#)), [Innatera](#), [BrainChip](#),

AI/ML Hardware Benchmarks

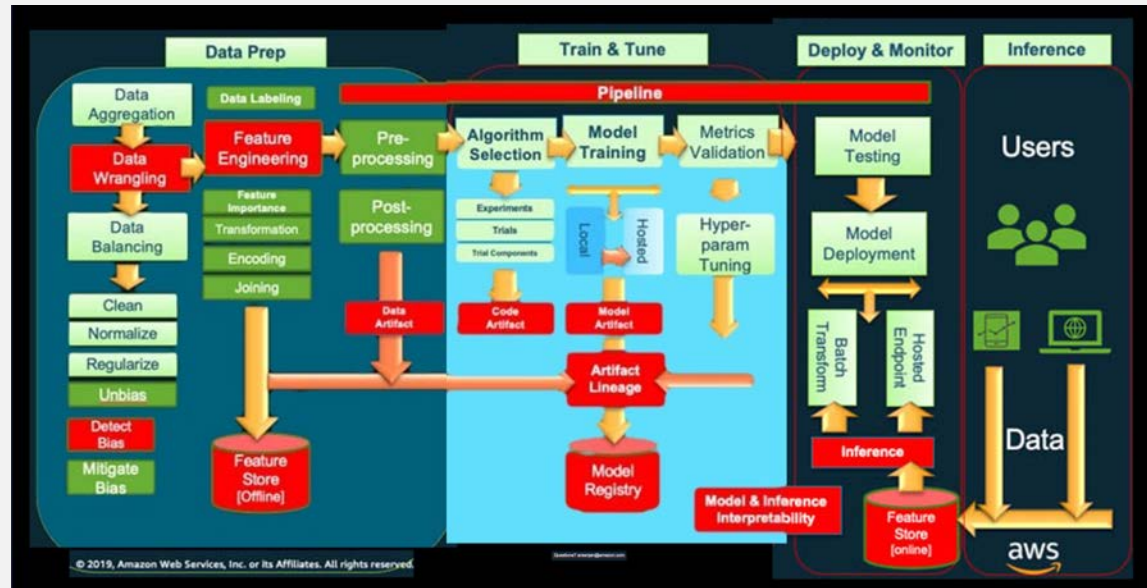
While there are lots of claims about how much faster each of these chips are for AI/ML there are now a set of standard benchmarks - [MLCommons](#). These benchmarks were created by Google, Baidu, Stanford, Harvard and U.C. Berkeley.

One Last Thing - Non-Nvidia AI Chips and the “Nvidia Software Moat”

New AI accelerator chips most deal with the software moat that Nvidia has built around their GPU’s. As popular AI applications and frameworks are built on [Nvidia CUDA](#) software platform, if new AI Accelerator vendors want to port these applications to their chips they have to build their own drivers, compiler, debugger, and other tools.

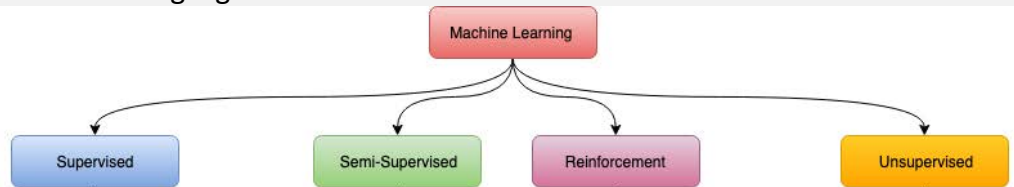
Details of a machine learning pipeline

This is a sample of the workflow (a pipeline) data scientists use to develop, deploy and maintain a machine learning model (see the detailed description [here](#).)



The Types of Machine Learning⁵ – skip this section if you want to believe it's magic.

Machine Learning algorithms fall into four classes:



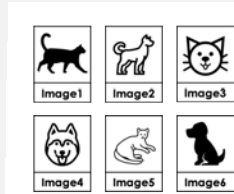
1. Supervised Learning
2. Unsupervised Learning
3. Semi-supervised Learning
4. Reinforcement Learning

They differ based on:

- What types of data their algorithms can work with
- For supervised and unsupervised learning, whether or not the training data is *labeled* or *unlabeled*
- How the system receives its data inputs

Supervised Learning

- A “supervisor” (a human or a software system) accurately *labels* each of the training data inputs with its correct associated output
- Note that pre-labeled data *is only required for the training data that the algorithm uses to train the AI mode*
- In operation in the inference phase the AI will be generating its own labels, the accuracy of which will depend on the AI’s training
- Supervised Learning can achieve extremely high performance, but they require very large, labeled datasets
- Using labeled inputs and outputs, the model can measure its accuracy and learn over time
- For images a rule of thumb is that the algorithm needs at least 5,000 labeled examples of each category in order to produce an AI model with decent performance
- In supervised learning, the algorithm “learns” from the training dataset by iteratively making predictions on the data and adjusting for the correct answer.
- While supervised learning models tend to be more accurate than unsupervised learning models, they require upfront human intervention to label the data appropriately.



Supervised Machine Learning - Categories and Examples:

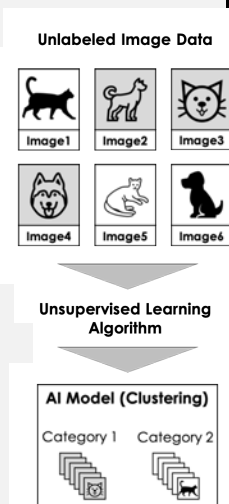
- *Classification problems* - use an algorithm to assign data into specific categories, such as separating apples from oranges. Or classify spam in a separate folder from your inbox. Linear classifiers, support vector machines, decision trees and [random forest](#) are all common types of classification algorithms.
- *Regression* - understands the relationship between dependent and independent variables. Helpful for predicting numerical values based on different data points, such as sales revenue projections for a given business. Some popular regression algorithms are linear regression, logistic regression and polynomial regression.
- Example algorithms include: Logistic Regression and the Back Propagation Neural Network

Unsupervised Learning

- These algorithms can analyze and cluster *unlabeled* data sets. They discover hidden patterns in data without the need for human intervention (hence, they are “unsupervised”)
- They can extract features from the data without a label for the results
- For an image classifier, an unsupervised algorithm would not identify the image as a “cat” or a “dog.” Instead, it would sort the training dataset into various groups based on their similarity
- Unsupervised Learning systems are often less predictable, but as unlabeled data is usually more available than labeled data, they are important
- Unsupervised algorithms are useful when developers want to understand their own datasets and see what properties might be useful in either developing automation or change operational practices and policies
- They still require some human intervention for validating the output

Unsupervised Machine Learning - Categories and Examples

- *Clustering* groups unlabeled data based on their similarities or differences. For example, K-means clustering algorithms assign similar data points into groups, where the K value represents the size of the grouping and granularity. This technique is helpful for market segmentation, image compression, etc.
- *Association* finds relationships between variables in a given dataset. These methods are frequently used for market basket analysis and recommendation engines, along the lines of “Customers Who Bought This Item Also Bought” recommendations.
- *Dimensionality reduction* is used when the number of features (or dimensions) in a given dataset is too high. It reduces the number of data inputs to a manageable size while also preserving the data integrity. Often, this technique is used in the preprocessing data stage, such as when autoencoders remove noise from visual data to improve picture quality.
- Example algorithms include: Apriori algorithm and K-Means



Difference between supervised and unsupervised learning

The main difference: Labeled data

- *Goals:* In supervised learning, the goal is to predict outcomes for new data. You know up front the type of results to expect. With an unsupervised learning algorithm, the goal is to get insights from large volumes of new data. The machine learning itself determines what is different or interesting from the dataset.
- *Applications:* Supervised learning models are ideal for spam detection, sentiment analysis, weather forecasting and pricing predictions, among other things. In contrast, unsupervised learning is a great fit for anomaly detection, recommendation engines, customer personas and medical imaging.
- *Complexity:* Supervised learning is a simple method for machine learning, typically calculated through the use of programs like R or Python. In unsupervised learning, you need powerful tools for working with large amounts of unclassified data. Unsupervised learning models are computationally complex because they need a large training set to produce intended outcomes.
- *Drawbacks:* Supervised learning models can be time-consuming to train, and the labels for input and output variables require expertise. Meanwhile, unsupervised learning methods can have wildly inaccurate results unless you have human intervention to validate the output variables.

Semi-Supervised Learning

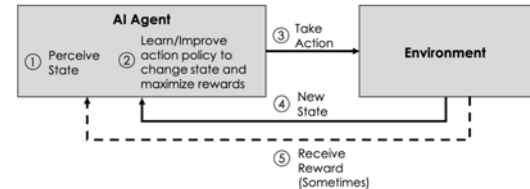
- “Semi- Supervised” algorithms combine techniques from Supervised and Unsupervised algorithms for applications with a small set of labeled data and a large set of unlabeled data.
- In practice, using them leads to exactly what you would expect, a mix of some of both of the strengths and weaknesses of Supervised and Unsupervised approaches

- Typical algorithms are extensions to other flexible methods that make assumptions about how to model the unlabeled data. An example is [Generative Adversarial Networks](#) trained on photographs can generate new photographs that look authentic to human observers (deep fakes)



Reinforcement Learning

- Training data is collected by an autonomous, self-directed AI agent as it perceives its environment and performs goal-directed actions
- The rewards are input data received by the AI agent when certain criteria are satisfied.
- These criteria are typically unknown to the agent at the start of training



- Rewards often contain only partial information. They don't signal which inputs were good or not
- The system is learning to take actions to maximize its receipt of cumulative rewards
- Reinforcement AI can defeat humans– in chess, Go...
- There are no labeled datasets for every possible move
- There is no assessment of whether it was a “good or bad move
- Instead, partial labels reveal the final outcome “win” or “lose”
- The algorithms explore the space of possible actions to learn the optimal set of rules for determining the best action that maximize wins

Reinforcement Machine Learning - Categories and Examples

- Algorithm examples include: [DQN](#) (Deep Q Network), DDPG (Deep Deterministic Policy Gradient), A3C (Asynchronous Advantage Actor-Critic Algorithm), NAF (Q-Learning with Normalized Advantage Functions), ...
- AlphaGo, a Reinforcement system played 4.9 million games of Go in 3 days against itself to learn how to play the game at a world-champion level
- Reinforcement is challenging to use in the real world, as the real world is not as heavily bounded as video games and time cannot be sped up in the real world
- There are consequences to failure in the real world



Sources:

- **Understanding AI Technology:** Greg Allen, Chief of Strategy and Communications Joint Artificial Intelligence Center (JAIC), Department of Defense

Visit our website:

www.vcarecrypto.com