Journal of Artificial Intelligence Research 79 (2024) 41-67 Submitted 08/2023; published 01/2024

The AI Race: Why Current Neural Network-based Architectures are a Poor Basis for Artificial General Intelligence

Jérémie Sublime JSUBLIME@ISEP.FR
ISEP – School of Digital Engineers,
LISITE Laboratory, DaSSIP Team, L303,
10 rue de Vanves, 92130 Issy-Les-Moulineaux, France

Abstract

Artificial General Intelligence is the idea that someday an hypothetical agent will arise from artificial intelligence (AI) progresses, and will surpass by far the brightest and most gifted human minds. This idea has been around since the early development of AI. Since then, scenarios on how such AI may behave towards humans have been the subject of many fictional and research works. This paper analyzes the current state of artificial intelligence progresses, and how the current AI race with the ever faster release of impressive new AI methods (that can deceive humans, outperform them at tasks we thought impossible to tackle by AI a mere decade ago, and that disrupt the job market) have raised concerns that Artificial General Intelligence (AGI) might be coming faster that we thought. In particular, we focus on 3 specific families of modern AIs to develop the idea that deep neural networks, which are the current backbone of nearly all artificial intelligence methods, are poor candidates for any AGI to arise due to their many limitations, and therefore that any threat coming from the recent AI race does not lie in AGI but in the limitations, uses, and lack of regulations of our current models and algorithms.

1. Introduction: AI Race and the Possibility of AGI

Artificial Intelligence (AI) has become an increasingly prominent field of research and development over the past few decades. In recent years, the development of AI has been accelerating, with more and more resources being devoted to the creation of increasingly sophisticated and capable AI systems that are now quickly released to the public and not contained to science and research activities anymore. In particular, the recent public release of *large language models* (LLMs) algorithms (Wei et al., 2022a; Bowman, 2023) based on GPT-3.5 (Brown et al., 2020), GPT-4 (OpenAI, 2023) and LLaMa (MetaAI, 2023) has taken the world by storm due to how we perceive that these methods are quickly progressing towards human-like conversation, writing and programming abilities, and how versatile they are in the type of problems they can solve. This is particularly striking for several models based on GPT-4 (Bubeck et al., 2023), such as Auto-GPT (Xaio, 2023), that can do complex conversation and programming tasks without much human intervention by simply directing instances of itself. Due to their coding abilities, their relative autonomy, and their apparently increasing intelligence, there are more and more claims that, without built-in or proprietary constraints, such programs could at some point modify and improve their own source code and reach the so called singularity (Vinge, 1993; Shanahan, 2015).

With more and more countries being in an AI arm race (Berg, 2023; Naughton, 2023), companies competing and racing to be the one with the next most disruptive and transformative AI system (Armstrong et al., 2016; Bostrom, 2017), or simply seeking to turn their idle data into assets (PwC, 2017), as well as eccentric billionaires also joining the race to develop their own version of those sophisticated tools to subdue their rivals (Lee, 2018; Castro et al., 2019), there is a growing concern (Maslej et al., 2023; Reich, 2023) among scientists, policymakers, and the public about the potential consequences of the so-called *AI race* (Han et al., 2020) and the role it may play in the emergence of AGI (Xiang, 2023).

AGI is an hypothetical superintelligent agent that could perform any intellectual task that a human being can, and do it faster and better due to its artificial nature. In other words, AGI could potentially surpass human intelligence in every way, from problem-solving to creativity. The fear of AGI mostly arises from the fact that such a system could be capable of self-improvement, leading to an exponential increase in its intelligence. This could potentially lead to an AI system that is far more intelligent than any human being, which in the general public conjures Hollywoodian scenarios of AGI that would completely escape our control, leaving mankind fate uncertain. Because our current AIs start to look like (in some aspects) the early versions of those we saw in fictions in doomsday scenarios that we imagined decades ago, they are generating unrealistic expectations and unnecessary fears (Cave & Dihal, 2019).

While there is no strict consensus, the following characteristics would be expected of an AGI (Legg & Hutter, 2007; Lake et al., 2017; Bubeck et al., 2023):

- It would need to be *general*: It could perform a wide range of intellectual tasks. This goes in opposition with the vast majority of current AI which are limited to a single or a handful of tasks.
- It would be *adaptable*: It would learn and adapt to new situations and tasks, rather than be limited to what it is pre-programmed for.
- It would be *creative*: AGI would need to be capable of generating new ideas and solutions, rather than simply following pre-existing rules or patterns.
- It would have *common sense* and *basic reasoning* abilities: It would need to be able to reason about the world in a way that reflects common sense understanding, rather than relying solely on hard statistical patterns or logical rules.
- It would need *long term memory*: By long term memory, we do not mean remembering data learned a long time ago, but that AGI should remember problems, tasks, steps or solutions that is was previously presented with, so that it doesn't make the same mistakes several times and has some *planning capabilities*.
- Optionally, *self-awareness* and *consciousness* are often discussed as required properties of AGI. However, it is unclear how to assess them for a computer program, nor if it is truly needed for intelligence.

Due to several recent algorithms edging closer to one or several of the previously described properties, both the claim that we are getting closer to AGI (Bubeck et al., 2023) and the fear of what may happen are rising again.

In this paper, we will review several of these recent and impressive AI algorithms, how they have evolved through time, and their architecture. In particular, we will focus on 3 categories of AIs: AIs developed to play games (checkers, go, Chess and Starcraft II), generative AIs (for artistic or deep fake applications), and personal assistant AIs built with large language models to chat with humans. Since they are all based on deep neural network technology, we will assess what this implies in terms of capacities but also limitations that such algorithms will have in the future. In particular, we will focus on reminding how these methods work and interface with the real world. By doing so, we will defend the argument that current deep learning based methods (that is, all current trendy AI methods) are poor candidates for developing AGI due to the inherent nature and many bottlenecks of this technology when it comes to learning or intelligence in general.

Our argument, will be a direct complement to the recent Microsoft paper on GPT-4 *showing sparks of AGI* (Bubeck et al., 2023), but not limited to the scope of LLMs model and with an additional architecture oriented perspective to the limits of current AI systems. We will finally develop on the idea that while they are not likely to become AGI, it does not mean that these algorithms or their future versions are harmless should their development be left unchecked in the current AI race.

Our paper is organized as follows: First, we will discuss some of the state-of-the-art AI that were introduced to the public in different fields, what they use in terms of deep learning techniques, and where they lie in terms of intelligence. We will follow with a section discussing the core architecture, principles and the limits of neural network-based models with examples from the two previous sections. Finally, we conclude our paper with the actual questions and dangers of neural network based AI systems and a discussion summarizing the different elements presented in this work.

2. AI has evolved fast and far: An Overview of the Current most Impressive AI Systems

In this section, we will present some of the most impressive AI systems to date in their respective fields and how they have evolved. We have sorted them in 3 categories:

- AIs for games,
- Generative AIs used for deep fakes or artistic purposes,
- Personal assistant AIs based on large language models.

Please note that these categories are arbitrary and were chosen for readability purposes and to better show their evolution in different domains of application. However, as we can see from Figure 1, they all come from the same family tree of methods and are overlapping in terms of technologies: For instance, the generative models (Goodfellow et al., 2014), which are the core of deep fakes and artistic AI, are also used by large language models which power the latest personal assistant AIs, and the adversarial learning process (Szegedy et al., 2014; Goodfellow et al., 2015) used by these methods is also used by the latest AIs for games. Finally, the vast majority of recent methods from all 3 categories all use deep neural network architectures.

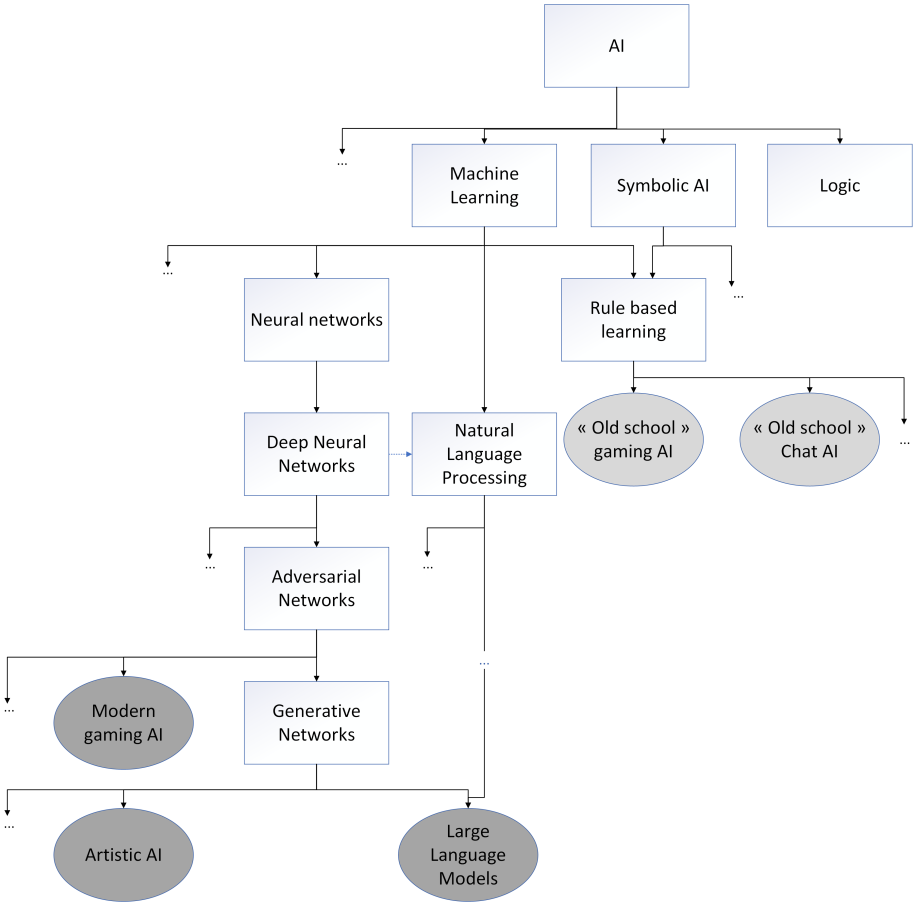

Figure 1: Simplified family tree of the AI systems presented in this paper. The families of methods discussed in this paper are displayed in grey. As we can see, modern gaming AIs, artistic AI as well as large language models all belong to the Deep neural network branch and don't have symbolic and rule-based AI in their ancestors. On the other hand, "old school" yet successful systems belong to the symbolic AI branch.

2.1 AIs for Games

AIs for games are the ones that speak the most for a broad non-scientific audience, and they also have been around for the longest, which is why we chose to start with them. Regardless of whether they were developed for traditional board games or for videos games, more or less advanced programs have been around since the 1950s to enable people to play alone, or rather against a machine: *Christopher Strachey* and *Dietrich Prinz* wrote computer programs able to play checkers and Chess respectively as early as 1951 at the University of Manchester. It is then *Arthur Samuel*'s checkers program, developed in the early 60s that was the first to be good enough to challenge an amateur player (Schaeffer, 2014). As for one player video games against adversaries, they started to appear with single player

games in the late 1960s and truly took off starting with 1978 "space invaders". However, the first occurrence of a computer program being better than human champions (without cheating) occurred with IBM Deep Blue Chess playing computer (IBM, 2008) who beat *Garry Kasparov* in 1997.

While Deep Blue appears to be the first great AI of our era, it is actually not a proper artificial intelligence program: It was an expert system that uses rules and logic, and had the capability of evaluating 200 million positions per second to search for the best next possible move. Deep Blue was trained using 700000 grand master games and by storing most of the possible Chess ending with 6 pieces, and all of them with 5 pieces or less. Using this huge database as a starting point, Deep Blue relied on alpha-beta pruning (Pearl, 1982) -an algorithm to make tree exploration more efficient- and its paralleled computation power to browse at high speed through the tree of possible moves to find the best one. In other words, Deep Blue was not really an artificial intelligence, but rather a brute force algorithm assisted by an efficient alpha-beta pruning method and huge hardware capabilities for its time. Still, Deep Blue has been a solid inspiration for later Chess playing AI such as Stockfish (Maharaj et al., 2022), which was also based on a pruning algorithm before being hybridized with neural networks, before this type of model culminated for Chess with engines such as Leela Chess Zero (a Chess adapted version of AlphaGO Zero that we will discuss in the next paragraph) (Silver et al., 2018). In Figure 1, all these models belong to the *"old school" gaming AI* branch that descends from symbolic AI and rule-based learning. This type of model has the huge advantage that it is explainable, and therefore we can understand why the algorithm chooses certain moves over others.

As we show in Table 1, with an average of only 20 to 40 moves per game and a maximum tree depth of possible moves estimated around 120, for the game of Chess alpha-beta pruning through the tree of possibilities and memorizing a large number of game to cover most possibilities remained a valid option. This is not the case however for the game of Go with its 19×19 board, 150 to 350 moves per game, and a tree depth as well as number of possible positions which are intractable. For this reason, the game of Go was long considered impossible to Master by traditional AIs that would simply try to brute-force through an intractable tree of possible moves. For the first time, a really smart computer program was needed, one that could plan its next move based not only on the branch most likely to lead to victory (which is impossible to assess given the depth of a Go game tree) but also based on the current situation and how to make the best of it.

To answer these new challenges, DeepMind technologies (a subdivision of Google) developed AlphaGo, the first AI able to play Go at high level and defeat world champions by combining neural networks and tree search (Silver et al., 2016). In its 2016 version that first beat World champion *Lee Sedol*, AlphaGo relied on this hybridization principles between a deep neural network and Monte Carlo Tree search. The main prowess resulted in training the neural network to help predict which branches of the tree (and consequently which moves) where the most interesting, without being able to browse to the full depth of said tree. AlphaGo and its successor can do so efficiently all the while being able to guess their likelihood of ultimately winning the game after each move. For its training, AlphaGo required a human database of 30 million of Go moves from 160000 games to attain a decent level, and was then further trained using reinforcement learning by playing against itself to

further increase its skills. Unlike all previous AI for board games, AlphaGo did not use any database of moves to play once the learning phase was over.

AlphaGo's successor by the same team, *AlphaGo Zero*, was an even more impressive case of artificial intelligence as it was trained without using any human games as reference, and only played against itself to figure out the best opening and moves (Silver et al., 2017). The development team reported that it took it approximately 15 million games against itself in a total of 3 days to reach a decent level, and 40 days and approximately further 200 million games against itself to be uncontested by both humans and earlier AI programs playing Go. It is because of its "self-training" abilities over such an enormous space of possibilities that AlphaGo Zero was seen as a major break through in AI. Furthermore, while the original AlphaGo was considered conservative in the way it played, this was not the case for AlphaGo Zero: it is also this "self-training" ability which led it to try moves and innovate during Go games in ways that humans would have never considered before. It was therefore able to surprise even the best humans champions due to the never seen before nature of some of its moves. Obviously, very much like Deep Blue in its time, all iterations of AlphaGo were supported by huge hardware capabilities, especially for the training phase of the neural network. In a way, we could say that while in the case of Deep Blue hardware was used to brute force a tree exploring algorithm, for AlphaGo Zero and its successor, the raw computation power was used to play a number of games so huge that the algorithm would self-train efficiently. Still, this shift in the use of computing power for modern AI proved useful as the technique used to train AlphaGo Zero was successfully reapplied to adapt it to the game of Chess and shogi (Silver et al., 2016), but also for protein folding problems (Heaven, 2020).

Moving to a new challenge, after AlphaGo Zero, the same DeepMind team turned to Starcraft II, an online real time strategy game by Blizzard Entertainement, as their new challenge to tackle with artificial Intelligence. Compared with Chess and Go, as can be seen in Table 1 the game of Starcraft II constitute a major challenge step up in the following ways:

- If we consider Starcraft II as a board game (which it is not), all games are played over maps that are more than 128×128 in size, which is way bigger than the 8×8 of Chess, or the 19×19 of Go.
- If you consider the number of mouse or keyboard actions as reference, an average Starcraft II game requires 700 to 3600 actions, which is a ten fold increase compared with Go.
- As a consequence of both the map size and number of actions, the number of possibilities and depth of a Starcraft II decision tree are considered to be infinite, which is also a step up compared with Go.
- While Chess and Go have time limitations, Starcraft II is not turn-by-turn and is played in real time.
- Starcraft II is way more complex than Chess or Go in the sense that a game requires you to do the following in parallel: develop your economy and gather resources, build units that will counter your opponent units (think rock-paper-scissor, but a lot more

complicated), control and send your units out to see what your opponent is doing, destroy his economy as well as his units.

- Battles between units go beyond which units are effective or not against another as unit micro-management is required in real time: for instance using units skills and spells at the right moments and places, but also moving and keeping damaged or fragile units in the back of your army whilst keeping tanky ones at the front.

- Starcraft II has *fog of war*, which means that unlike Chess and Go, you can't see what your opponent is doing unless you send troops out to see what is going on.

From this description it is easy to see that to be good at this game without cheating (no fog of war, more powerful units, increased resources, etc.) an AI would need to have many skills that resemble what would be needed for an AGI: planning capabilities, memory of what happened a few moments ago, real time response abilities, and creativity.

Due to the large exploration space for a game like Starcraft II, a full self-training was not possible, and AlphaStar was pre-trained using a database of 65000 games to learn the very basic moves and some basic strategies. It is only then that it used the same self-training ability by playing against itself to refine its strategies. This reinforcement learning phase also included a phase against so-called *exploiter agents* whose purpose was to tackle the main agent on its own weaknesses so that it could improve.

Game	Deep Blue Chess	AlphaGo Go	AlphaGo Zero Go	AlphaStar Starcraft II
Year	1997	2016	2017	2019
Board size	8×8	19×19	19×19	$> 128 \times 128$
Average lowest number of moves	20	150	150	700[1]
Average highest number of moves	40	350	350	3600[1]
Number of possible positions	$10e120$	Intractable	Intractable	Infinite
Tree depth	120	Intractable	Intractable	Infinite
Training games	700000	> 160000[2]	196000000	365000[3]
Human equivalent training time (years)	19.98[4]	37+[2,5]	44750[5]	200[3]
Real time game ? [6]	NO	NO	NO	YES
TPU or equivalent	≈ 27[7]	48	4	16
Trained from human games ?	YES	YES	NO	YES
Able to self-train ?	NO	YES	YES	YES

Table 1: Comparison of 3 famous gaming algorithms in terms of complexity of their final application and their models: [1] computed on the basis of 70 to 120 actions per minutes for games lasting 10 to 30 minutes; [2] The number of self-training games is unknown; [3] The original algorithm was trained from 1000 pro games, but DeepMind declared 200 years equivalent of human training, and we estimate that pro-gamers play on average 5 games a day; [4] Assuming 15min games; [5] Counting 2h per game on average; [6] While Chess and Go are played with time limitations, Starcraft II requires real time actions and responses; [7] Estimated using 0.42 GFLOPs for a TPUv3, knowing that Deep Blue was estimated at 11.38 GFLOPs.

Very much like for AlphaGo, world top Starcraft II players that faced AlphaStar were surprised to see the algorithm using strategies that had never before been seen in human games. Also like for the game of Go, the AI quickly proved to be able to defeat any human player. Furthermore, due to the fact that it was initially unlimited in the number of actions per second it could make, it was nearly impossible for a human to beat AlphaStar's unit micro-management, as humans are limited by their physical abilities as well as their use of mouse and keyboard, and can't sustain a very high number of actions per minutes for long (typically 60 to 120 actions per minute at best). For this reason, AlphaStar was capped in terms of number of actions per minutes it could make, but still proved better than human players. However, it is worth mentioning that unlike for the game of Go, there were several instances where AlphaStar showed its limits by being unable to assess a situation it had probably never seen before or had an erratic play style that made no sense at all: For instance, Starcraft II uses a wide array of maps (boards) that are all different and 3 different races. AlphaStar was initially able to play a single race and was limited to the sets of maps it knew, and was unable to adapt to others. While switching race can prove challenging for beginner humans players, it does not result in "nothing happening". As for map changing, it presents no problem at all for humans (even at very low level) and shows that humans have an adaptability that it does not have. Finally, in some matches, AlphaStar proved unable to properly assess whether it was losing or winning, which resulted in erratic behavior very similar to what could often be seen from an "old school" AI going out of its decision tree.

It is worth mentioning that while attempts have been made at reproducing AlphaStar-like performances with smaller networks and less hardware (Liu et al., 2022), such feats are so far limited to companies with huge research departments and computation power.

2.2 Generative or "Artistic" AIs

In this paper, we call *artistic AI* the different algorithms that have been released with the ability to create artistic or realistic images of videos on different subjects. This type of AI first appeared with the neural network called Inception (Szegedy et al., 2015) whose original goal was to detect objects in images, but was later used as a reference to understand how convolutional neural networks worked, and in particular their different convolution layers. Using the Inception network, a team from Google proposed the *DeepDream* software, a program generating psychedelic and dream like images. While still used to detect elements of interest in images (mostly faces), DeepDream uses a reverse process compared with normal detection network and will twist and adjust the image to look like something else by giving an output neuron more importance than it should have (a cat face instead of a human face for exemple) and will then proceed to alter the original image via gradient descent so that it matches with the purposefully wrongly activated neurons. This results in very strange images reminiscent of what can be experienced by LSD users, which led some scientist to believe that convolutional neural network share common architecture with the visual cortex of humans (Schartner & Timmermann, 2020).

While DeepDream was the first neural network to generate false images, it is later networks based on generative adversarial networks that really became known to the public audience for their ability to generate the so called *deep fakes*: images or videos artificially

generated and for which it is very difficult to say whether it is a real picture, or something generated by an AI.

These deep fake networks use both the autoencoder principle (Hinton & Zemel, 1993; Kingma & Welling, 2014), (See next section and Figure 3 for details): the encoder finds a lower space representation of a person or thing to modify. Once this is done, key features can easily be changed in the reduced feature space, and the decoder will proceed with the reconstruction of the modified (fake) image. Coupled with a generative adversarial network (Goodfellow et al., 2014) after the decoder, this technology has proven to be very powerful. The main principle of generative adversarial network is to have 2 networks following an optimization process against one another:

- The first network (the discriminator) is trained to detect real images or videos from artificially generated ones.
- The other network (called the generator) tries to generate images or videos that won't be detected as fake by the first network. It is optimized to make fakes that are more and more difficult to detect.

These deep fakes can be used in all sorts of ways: adding, replacing or removing something or someone from an image, changing small or major details, generating a video of someone saying a speech that never happened, building a full composition image of things that never happened (for example in image of the pope partying in Coachella), etc. Because of the adversarial training process, very realistic generators can be trained, and their generated images are impossible to tell from real ones.

While not a proof of general intelligence per se, these deep fake algorithms can generate from scratch very realistic images on demand by simply having a user describing what he or she wants. It can mimic any style of photo, painting or art, and there is also the possibility of using random parameters to generate images or videos. In any case, with the best of these generative algorithms, the output will be very realistic and unique, which can be interpreted as a form of creativity.

2.3 Personal Assistant AIs

We will now move on to the last category of AI discussed in this paper, the so-called "*assistant AI*". This category regroups all AIs models that were developed to interact with humans with the goal of helping them: Personal home assistant such as Amazon's Alexa, Google Home, Microsoft's Cortana, and Apple's Siri, the many chatbots developed for support services, and obviously conversational AIs such as ChatGPT.

Conversation has always been considered a difficult task as it requires to both understand a question, and producing an answer that is accurate and understandable. AIs targeted at this type of task therefore have to master language data, which have some very specific challenges:

- Words are more than simple data that can be hard encoded into binary vectors. An efficient AI algorithm would have to learn (or at least use) a word embedding system in which closely related and semantically similar words are similar enough in the embedding space. This is what Word2vec (Mikolov et al., 2013) does for instance.

- Understanding language requires more than knowing a list of words and their semantic relationships: it also entails a basic understanding of grammar in order to properly understand a question, a prompt or a text: the tense of a verb, the presence of negations, and the grammatical role of some groups of words can completely change the meaning of a sentence.

- Most languages contain at least 80000 to 300000 words, which is a lot to learn, especially if you add the semantic. Grammar also varies from one language to another, although there are some common basis.

- Specific idioms and second degree humor can make a sentence all the more difficult to understand.

All the reasons mentioned above have made assistant AI particularly difficult to develop, and it is only recently that the first effective assistant AIs have appeared. Indeed, before the generalization of neural networks to learn large corpuses and semantic relationships, most assistant AIs simply relied on the detection of a limited number of keywords and expressions, and proposed pre-set answers accordingly. Most early chatbot fall under this category and -very much like early gaming AIs- belong to the family of *Symbolic AI* shown in Figure 1. Assistants such as Google Home or Alexa use neural networks to process the sounds they hear into instructions with words given to their algorithm. But the "reasoning part" to interpret the instructions is similar the early chatbots and also belongs to the symbolic AI branch to decide what to do or to answer.

As one can see, the main difficulty in natural language processing is not so much the decision making or the generative process, it is to properly model the language to both understand and answer questions. Once again, it is deep neural networks that brought the main advances in the field of natural language processing. Word2vec (Mikolov et al., 2013) is one of the first proposed word embedding system that accounts for word semantic in text data representation. It was more recently followed by BERT (Devlin et al., 2019), another embedding system relying on neural networks. The main difference between the 2 technologies are the following: BERT allows several vector representation for the same word, while Word2vec only has a 1-to-1 mapping. BERT can handle words that are outside the original vocabulary it was originally trained with, while on the other hand Word2vec cannot do so. From Word2vec and BERT, the next main advance was large language models (LLMs) whose principle is to take as input a list of tokenized words using embedding (the question or prompt) and to output a probability distribution over the vocabulary known by the system and which will be used to build an the answer. These systems are called "*large*" because they use billion of parameters and are trained over very large bases of vocabulary. With 340 million parameters and a corpus of 3.3 million words, BERT is considered to be the first large language model, and is somewhat small compared to GPT-3 (175 billion parameters and 300 billion tokens), LLaMa (MetaAI, 2023) (65 billion parameters and 1.4 trillion words) or GPT-4 (OpenAI, 2023) with more than a trillion parameters.

AI systems belonging to the LLM family have shown rapid and impressive progresses:

- With each new iteration or version, they act more and more human in the way they interact with people and can do casual conversation and have the apparent ability to solve logic problems just as well as humans do.

- They are fluent in a large number of human languages and can translate easily from one language to another.
- In addition to natural languages, many of these AIs now have programming ability.

In terms of technology, these systems cover two main tasks:

- Learning and embedding the large set of words or tokens that constitute their vocabulary. We have already discussed this difficulty and this is done by pre-training them on very large textual databases such as Wikipedia or GitHub.
- Training the neural network to actually produce good quality answers based on an almost infinite possible number of prompts. This is done using a mix of self-supervised learning and reinforcement learning from both human and other AI instances. This is very similar to what we have seen for AlphaGoZero and AlphaStar, but applied to language.

These systems have recently surprised the World with the ability of some of them to successfully pass the Turing test (and successfully pretend to be human), the large panel of tasks they can do with efficiency, but also some very specific abilities that appear to have spontaneously emerged in some of them without being explicitly programmed.

Most recent LLMs also appear to have shown some reasoning abilities. However, these reasoning abilities seem to be highly dependent on breaking-down or splitting the problem into simpler sub-problems with intermediate prompts to increase the likelihood of a good understanding by the neural network (Wei et al., 2022b; Zhou et al., 2023). Furthermore, it is very difficult to tell the difference between reasoning abilities and just having learned on a database large enough to find the answer without any reasoning, and some researchers even doubt that it can reason at all due to its lack of world model (Borji, 2023).

Indeed, it is worth mentioning that they have also shown limits, such as the so-called *hallucinations* in which these systems proposed with great certainty an answer to a given query which is not correct and sometimes has no real basis. A good example of such hallucinations is ChatGPT proposing references of scientific papers that do not exist: the journal exists, the authors have realistic names, the title seems to match the question asked to the algorithm and would be relevant for the journal, but the paper does not exist. This is due to the way these algorithms are programmed to find the most likely distribution in their known vocabulary to propose an answer: with most questions, it just takes putting these words or tokens in a certain order to form a fine sentence and a correct answer. Doing so with pieces of a scientific paper and a matching journal will however most likely result in something that does not exist. The same kind of strange behavior can happen when asking GPT-4 for the names of the feodal lords of some lesser-known villages, which it will answer by making up realistic names that are out of touch with the reality. There is another famous example where GPT-3.5 was asked to remind the places of a series of past conferences, and it only returned the ones for odd years. When asked why he did not return the places for even years, it answered that the conference did not take place on even years (which was false). Likewise algorithms such as LaMDA (Thoppilan et al., 2022) have been confused by user prompts asking if Yuri Gagarin went to the Moon (which the algorithm successfully answered no), before affirming that he brought back moon rocks that he got from the Moon.

These four examples show the limits of these systems which mimic intelligence but are actually mostly trying to give you the most likely answer, which is different from the truth.

Finally, we can also mention that some "spontaneous" behaviors of such AI may sometime be very inappropriate: Microsoft Bing AI has had a few bad experiences with chatbots going horribly wrong: The Tay chatbot in 2016 was a first example of an AI that went nazi very quickly because it was fed with the wrong data by malicious users. And more recently some of their latest LLM AI also acted very creepy with some of their users by either faking sentiments for them, or displaying downright hostility.

3. State of the Art AI Systems and their Deep Learning Limitations to Evolving AGI

In this section, we discuss the building blocks, architecture, and training methods of the neural networks that are at the heart of most modern AI systems nowadays. In particular, we explain how these common blocks used in all modern neural networks are defining the capabilities and limitations of modern AI systems.

It is worth mentioning that the Deep Learning paradigm is very likely to remain the dominant AI technology for the decades to come, and that as such everything presented in this section is valid for current AI systems and for most systems to come. Indeed, as shown in Figure 1, all breaking ground new AIs systems come from the same family descending from Machine Learning, deep neural networks, adversarial networks, and more recently generative networks.

The main implication of all these algorithms coming from the same families of methods is that they have many things in common as they inherit the strengths, weaknesses and core principles of the same methods.

3.1 Core Principles and Main Components of Deep Neural Networks

Artificial neural networks are a family of machine learning methods remotely inspired from the neurons in the brain, and whose earliest apparition in computer science was in the 1960s with the first perceptron (Minsky & Papert, 1969) and the conceptualization of backpropagation for optimization purposes (Amari, 1967). The field then stagnated until the late 1980s with the first conceptualization of neural networks able to learn phonemes (Waibel et al., 1989), and the idea of convolutional neural networks inspired by the human visual cortex for image analysis and interpretation using artificial intelligence (LeCun et al., 1989; Rumelhart et al., 1986). The field then rose to popularity and emerged for the general public in the early 2010s when GPU-based computation power became available and cheap enough to allow for larger and more sophisticated networks to emerge and be trained faster.

In Figure 2 we show the basic components of all neural networks. First, we have neural units which basically process a set of inputs which are multiplied by weights to be learned. Each unit also contains an internal activation function which will determine the output based on the weighted sum of the neural unit inputs. A neural network itself is made of several layers containing such neural units. We distinguish between different types of layers:

- Input layers which handle either raw input data, or outputs coming from earlier blocks of the networks, and transform them into something usable by the rest of the

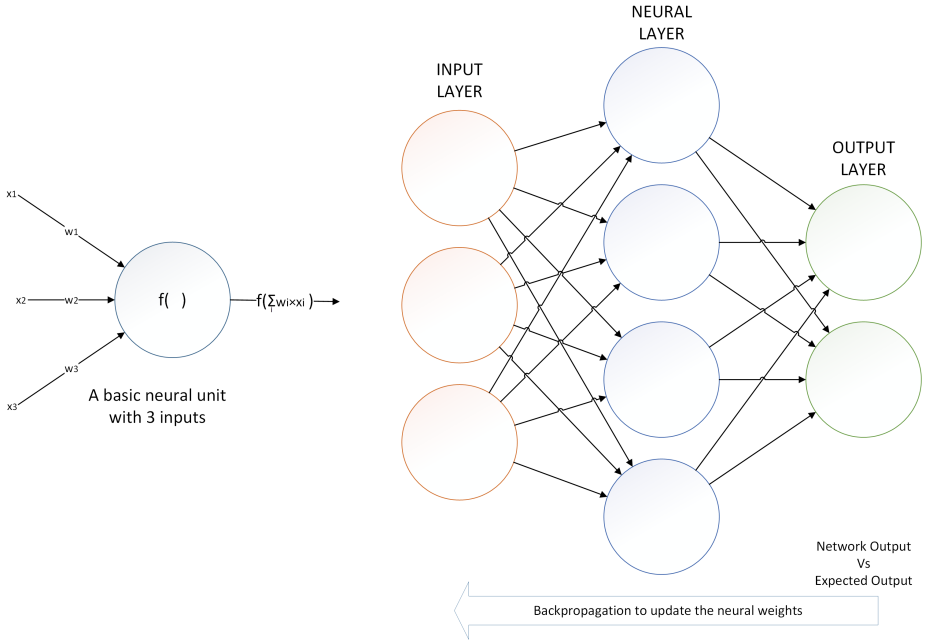

Figure 2: On the left: a basic neural unit with 3 weighted inputs. - On the right: a simple network with an input layer, an output layer and a single neural layer in the middle.

network. For instance, text data needs to be transformed into numerical vectors, and convolutional layers are typically used with images to extract features at different scales.

- Neural layers between the input and the output. A typical neural network may count dozen or even thousands (hence the name *deep* neural network) of these layers. They are used to find and disentangle complex representations of the input data.

- Output layers which basically produce the answer of the network. This answer can be in the form of a numerical vector, in which case these input layers are not very different from typical neural units. However, in the case of a classification problem, it is typically expected that only one neuron from the output layer activates to decide which class is picked. This required specific activation functions and layer properties.

Neural networks are trained by means of gradient descent and back-propagation: The networks (or blocks of the networks) are fed with multiple data which are going to run through the different layers and produce an output. This output is then compared with what was desired for the data that were fed, and the difference between the produced result and the expected one is then back-propagated over the different layers of the network(s) so that the weights of the different units are adjusted to better match the expected result. It is worth mentioning that it is typical to use the root mean squared error, and not the

raw difference, but that many different functions can be used depending on the task and problem.

The goal of any neural network training is therefore to optimize an objective function so that the network output becomes satisfying enough. A neural network such as the one shown in Figure 2 is very basic, but more complex neural networks follow the same basic architecture with more layers, layers containing more units, and complex structures containing several sub-networks filling different functions.

3.2 Interfacing to the Real World and Extracting Features: A First Bottleneck

As we have seen with Figure 2, no matter their complexity or depth, all modern AI systems interface with the real world using an input and an output layer: Gaming AIs take the current state of the game (or what they can see of it) as input, and output their next move. Personal assistant AIs take a written or oral prompt as input, and will output text or spoken answer accordingly. Artistic AIs will take a set of parameters as input (type and size of image, a theme, or even a written description) and will output an image. Etc.

In most fiction works where an AI becomes aware and escapes, it starts to connect to systems it was not supposed to connect to, and to do things it was not meant to. This is absolutely impossible due to the fixed and static nature of the input and output interfaces of neural networks: The way neural networks are trained, which we have already briefly mentioned, and will detail more in the next sub-section, implies that the structure of the input and output layers of any neural network is fixed. If it changes or is modified, the training must basically be done all over again from scratch.

Beyond simple interfacing limitations restricting the type of input and output, we can also mention task specific interfacing limitations:

- For instance, we have seen with text analysis tasks that there were different ways to feed text data to an AI using Word2vec, BERT or large language models. In any case, words or languages unknown to the algorithm would be difficult to understand for any AI system as they would not be included in their original embedding system.

- In image processing, where convolution layers are key to analyse an image, it does not take the same architecture to make a classification method to tell cats from dogs, and to analyse medical images: The convolution layers needed to search for specific elements at different scales would simply not be the same. It means in this case, that even if an AI can interface to the 2 types of images, it may not have the right first layers to properly extract the relevant features.

Indeed, while we have simplified things by only discussing the first input layer as the only interface with the external world for an AI, in truth while the first layers limit the format of what can be accepted or not, there is a large number of the first layers which is used to create the features that will later be processed by follow-up layers in a process known as *feature extraction*. This means that it is not only the format of the input layer, but also the format of the follow-up layers for feature extraction (e.g.: number and type of convolutions) which will determine what an AI efficiently interfaces to or not, even before it has been trained. And once the training has been done for these feature extracting layers, it will narrow-down the possibilities even more.

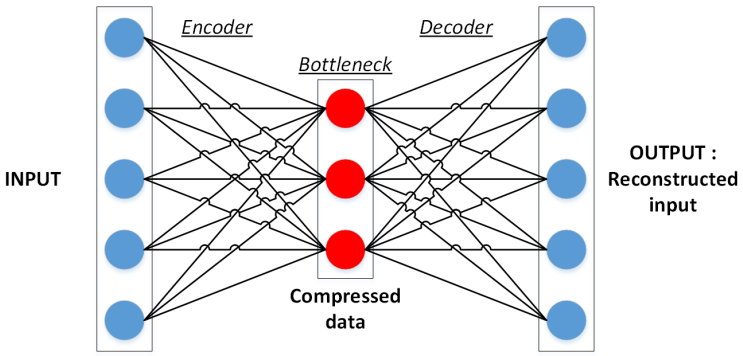

Figure 3: Example of a very basic autoencoder with a single hidden layer for compression.

In Figure 3, we show a typical autoencoder (AE) architecture (Hinton & Zemel, 1993), an unsupervised model used for feature extraction and whose principle is to train a network to rebuild its input with a certain number of compression layers in between. The first part before the narrowest bottleneck layer is called the encoder, and the one from the bottleneck to the reconstructed output is the decoder. Once trained, the decoder part is discarded, and the encoder can be kept and integrated into a larger neural network to transform original input data into better quality features. Yet, these "better quality features" would also be fixed by the encoder architecture and training, and would be just as much a limitation as the input layer. Once again, modifying them on the fly by adding or removing layers or neurons is usually not an option. Variational autoencoders (VAEs) (Kingma & Welling, 2014) are an evolved form of autoencoders that maps the data into a latent space and inject white gaussian noise between the latent space and the decoder to achieve more robust features. AEs and VAEs otherwise fit the same role of an unsupervised and self-supervised learning of the features.

To sum up these first sections on the limitations of current neural network based AI, we have seen that the input and output layers, as well as feature extraction mechanisms which enables these AIs to connect and interact with the outside worlds are limited and rigid. Furthermore, while we have simplified the problem to simple networks, in large AI systems which are made of very large deep and multi-component sub-networks, these structures are everywhere in the network, which further limits its adaptability to efficiently interface with anything else than what it was designed for. Finally, while they are simpler, the output possibilities of such networks are perhaps even more limited in their format.

3.3 Limits of Objective Functions and Gradient-based Learning

Let us now discuss objective function based learning and gradient descent which are the two core processes of Deep Learning algorithms. We will see how and why this type of learning is not compatible with the idea of AGI because of the way they restrict the types of tasks that can be handled and are also currently incompatible with symbolic learning (they don't belong to the same AI branch as shown in Figure 1), a key feature of human-like learning.

The first obvious limit of objective based-learning is that some problems are not easy or convenient to model with an objective function: While it is somewhat simple to have

a quality-based objective function for classification problems, or a reconstruction error to minimize for regression methods or autoencoders, it is more difficult for complex tasks. For instance, we mentioned earlier systems for games or personal assistants. In this case rating a game move or a proposed answer and turning it into an objective function when the"*best*" move or answer are not known can be very challenging: Can we define such function when the best answers are not known ? And will it be possible to differentiate it and run a gradient descent ?

This leads us to our second argument on the limitations of objective-based and gradient based learning: it implies that the objective function must be differentiable. Indeed, the backpropagation system used by deep learning methods requires that a derivative can be found in all layers, including the output one with the objective function. Furthermore, in the case of a non-convex system (which is very often the case), convergence towards a local minimum rather than the optimal one is very likely with gradient-based learning, which is another limitation.

Lastly, in its current form, both gradient-based learning, objective-function based learning and the limited input interfaces are incompatible with symbolic-based learning which encompasses structured data and is very useful to insert human-made rules into an AI system, but also to understand the decisions made by an AI. There is no place in gradient-based learning for hard rules (only for targeted examples to guide a system), and even less for complex structured data.

To sum up this subsection, objective function and gradient-based learning are limiting in two ways:

- First, they cannot handled symbolic learning and structured data.
- Finding the right and differentiable objective function can be challenging and is far from ideal: Even if one can be found, it will necessarily limit the type of tasks that an AI can process or not, and the risk of convergence towards a local optimum is always a risk.

3.4 Learning Processes: More Bottlenecks

Let us now focus on the learning abilities of the current AI model. Our goal in this subsection will be to show that current AI models cannot lead to an explosion of generalized intelligence able to answer or to know everything. To do so, we will use geometry to illustrate the problem. Let us represent the space of problems an AI should be able to answer in its specific domain as a line in space:

- If the problem is simple and finite, the line is also finite.
- For complex problems such as generalized AI, the line is most likely infinite.

The way most current AI methods based on deep learning work is that they are fed with examples belonging to the space of possible problems (LeCun, 2018): If the examples are provided with correct answers, then we are dealing with *supervised learning*. If the algorithm is provided with just examples, left to explore and rewarded or penalized depending on its answer, then it is *reinforcement learning*. If the algorithm is only fed unlabeled examples

and left on its own to provide answers in an unsupervised way, then we are dealing with *unsupervised learning*. The type of learning does not really matter for our analysis:

- Before its training, the algorithm will have been provided with examples covering some of the considered problem space.
- Once it is trained, the algorithm should be able to properly answer a certain amount of problems from the same space.

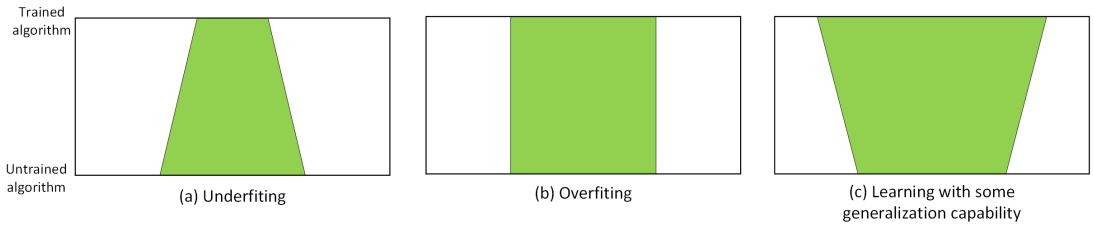

Figure 4: Geometric representation of underfiting, overfiting and fair generalization capabilities: The bottom and top lines of each rectangle are the problem space. The learning process from bottom to top is in green and show how much of the problem space can properly be dealt with after the training based on the space covered by the training examples.

There are 3 possibles outcomes to training an AI: In the first scenario, the AI underfits and is only able to handle a much narrower number of cases (sometimes 0) than what was fed to it during its training. This usually means, that the AI structure and models are not complex enough to properly grasp the problem, or when the number of examples is too small compared with the algorithm complexity. In the second scenario, the AI overfits during the learning process and will only be good at processing examples it has already seen, but bad with anything else. This can be caused by all sorts of sampling issues with the training examples, but also by an algorithm structure too complex compared with the problem complexity. In the last scenario -which is the prefered one-, once it has been trained the AI algorithm can be effective at handling problems from a space larger than the set of examples it was fed. As hinted by the two previous scenarios, this can be achieved only with the right complexity for the AI algorithm relative to the problem, a sufficient number of examples also relative to the complexity, and also a good enough sampling.

These 3 scenarios are visually shown in Figure 4, where the bottom and top lines of each rectangle are the problem space and the learning process from bottom to top is shown in green. In this representation, and based on the previous explanations, we see that the learning capability of an algorithm can be modeled as a function $\mathcal{L}(C_X, S_X, C_A, S_A)$ which produces a result which will reflect on whether or not the algorithms will be able to tackle data beyond what it was fed:

- This result will be negative if the algorithm turns out being unable to produce good quality results even on the data it was trained from. This is known as *underfiting*.

- The result will be 0 in case of an *overfiting*. That is, after training the algorithm is efficient on the data it was trained with, but ineffective with anything else.

- A positive result greater than 0 if the algorithm shows *generalization capabilities* after its training and can efficiently process data it had never seen previously. Obviously, the greater the number, the larger the generalization capability.

In a way, this function would define an "angle of some sort" which is illustrated in Figure 4 by the problem space where the algorithm is efficient being narrower, the same, or larger, after training.

This function would have many parameters including C_X the overall complexity of the problem, S_X the size and representativity of the set of training examples relative to the problem complexity, C_A the complexity of the AI model, S_A the size of the set of training examples relative to the complexity of the AI model. There are very few known properties for such function:

$$\forall C_A : \quad \frac{\partial \mathcal{L}}{\partial C_X} \leq 0, \quad \frac{\partial \mathcal{L}}{\partial S_X} \geq 0, \quad \frac{\partial \mathcal{L}}{\partial S_A} \geq 0, \quad \lim_{S_A \to +\infty} \mathcal{L} = 0, \quad \lim_{S_X \to +\infty} \mathcal{L} = 0 \tag{1}$$

Based on this first idea, let us now add the interfacing constraints that we have discussed previously: Deep learning algorithms have fixed input layers which restricts what they can and cannot efficiently process. In our geometric representation, this gives hard limits such as the ones shown in Figure 5(a) where no matter the quality of the algorithm, the learning is restricted by the narrow interfacing possibilities.

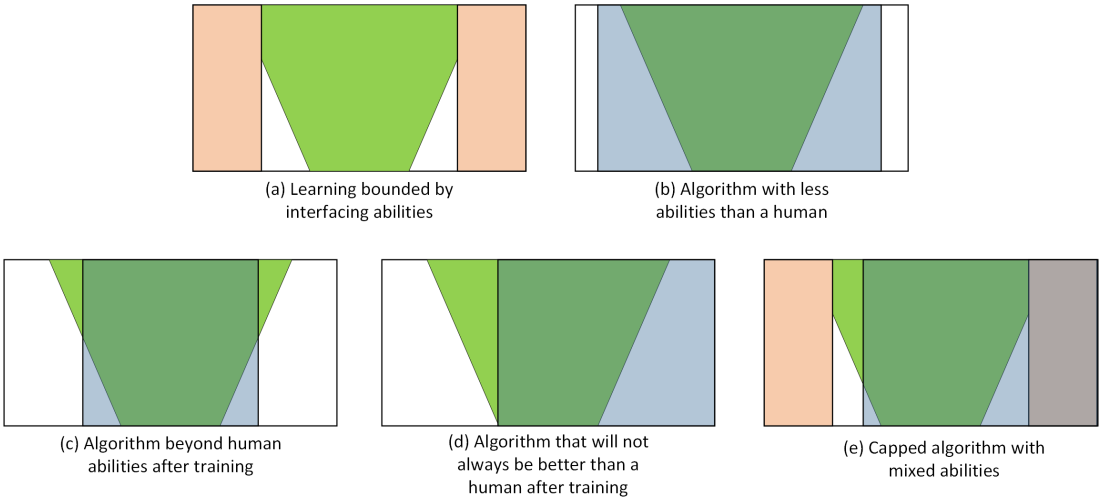

Figure 5: Geometric representation of different scenarios: The bottom and top lines of each rectangle are the problem space. The learning process from bottom to top is in green, the hard interface boundaries are shown in red, and the boundaries of reasonable human efficiency are in blue.

If we keep looking at Figure 5, we see a couple of other useful examples to understand the strengths and limits of AI methods. In this Figure, we show in blue the boundaries of *reasonable human efficiency* over the same application domain than the algorithm. Please note that the algorithm being outside of the *reasonable human efficiency* zone does not necessary mean that it is smarter. Indeed, many of the examples outside of this zone may well be non-sensical, and there is no benefit in having the algorithm able to process them.

The goal of many Machine learning algorithms and AI is to be as effective or more effective than humans, and to do so with minimal training efforts. This would mean having a narrow number of examples compared with the global human efficiency area to train the algorithm, and an equal or larger efficiency of the AI after the training. This ideal scenario is shown in Figure 5(c) where the green area is narrower than the blue one at the bottom of the rectangle, but larger at the top. This also implies $\mathcal{L}(C_X, S_X, C_A, S_A) > 0$ as a necessary, but not sufficient condition. Indeed, Figure 5(b) shows the very common example where despite having good learning properties, the AI remains less versatile and efficient than a human on a given domain after its training. Figures 5(d) and 5(e) show two other likely scenarios: In the first one the AI exceeds human capabilities for some part of the domain problem, but remains inferior in others. In the second one (which is even more common), we have the same phenomenon with an additional area of human efficiency which will never be reached by the AI because of its interface limitations.

Likewise, we have seen with large language models, that they require humongous amounts of data crawled from the internet and that they do not even guarantee good results. Both families of algorithms tackle this issue by having AI agents training with or against one another to further improve their performance in a process called adversarial learning that we have presented already. This process makes it possible to have, not one, but several rounds of stacked training with the same AI architecture. If such AI is good enough at the early stage (it has a positive learning angle), it will lead to proficiency in a potentially larger number of training examples after each iteration of the training, which can be used to further improve the AI performances. We have seen how strong this process of stacked reinforcement learning and adversarial learning can be for gaming AI and large language models. It is part of what has misled some into thinking that such process may lead to an explosion of intelligence and ultimately to AGI.

In Figure 6, we show why this idea is false with 3 scenarios that take their explanations from the properties of Equation (1):

1. In the first scenario from Figure 6(a), we see how the first pre-training starts with a narrow set of examples and a narrow yet positive angle which results in a much better learning angle with more examples at the second step of learning. Then because of properties (4) and (5), we know that for a fixed architecture and problem the learning angle is bound to go towards zero as the number of training examples increases. This means, that regardless of the number of incremental learning iterations, improvement will decrease and stop at some point.

2. Scenario 6(b) shows that we can't even be sure that there will be an increase in learning rate at all between the different learning steps. It is just as likely to get an ever decreasing angle towards 0.

3. Finally, Figure 6(c) shows a last scenario where, regardless of the learning angle, it is the interface limitations of the algorithms which will stop the progress of the AI before it reaches its sample size limits.

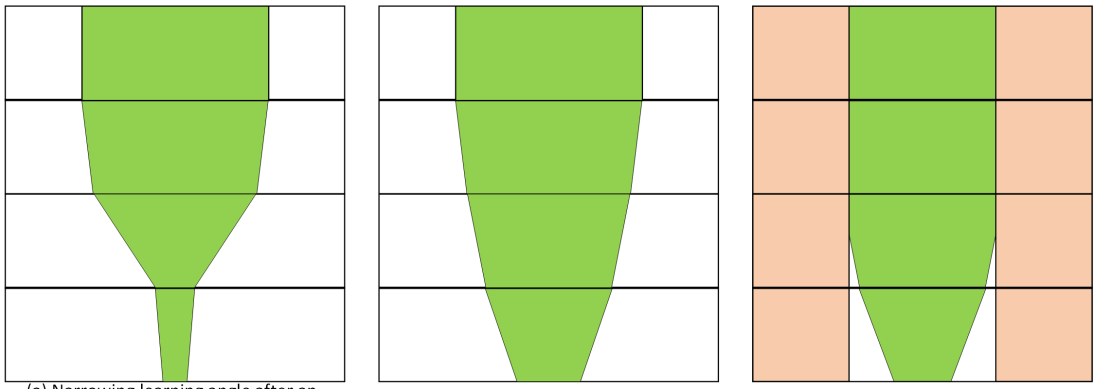

Figure 6: Geometric representation of different scenarios for stacked reinforcement learning: The bottom and top lines of each rectangle are the problem space. The learning process from bottom to top is in green, and the hard interface boundaries are shown in red.

While it is tempting to say that all it takes to remove these limits is to change the algorithm architecture or input layers between training phases, we remind our reader that this is not that easy:

- Even though knowledge acquired during the training process with an earlier architecture could be re-used, any architecture modification will result in a new network that has to be re-trained from scratch. In other words: back to the first phase of training.
- Modifying the input interfaces will result not only in having to retrain the AI from scratch, but also in having to redefine training samples entirely.

4. Conclusion

4.1 Limitations

In this paper, we have studied several aspects that AI would need to achieve to become what an AGI, and how current state of the art systems are both coming close to some of them, but also falling short for many of these aspects: The idea of being *general* and able to tackle a wide variety of problems by contrast to a narrow range of problems, the notions of adaptability and creativity, common-sense and basic reasoning abilities, and finally long term memory.

Through 3 different types of AIs (AIs for games, generative artistic AIs and personal assistant), we have seen how each family has modern AI that seems to check several of the

previously mentioned criteria: Gaming AIs have creative and reasoning abilities as well as long term memory abilities that far surpass these of humans. Generative AIs are also able to display creative abilities that result in new and unique pieces of art. And finally, personal assistant AIs based on large language models have shown long term memory, adaptability, creativity, common sense and basic reasoning abilities, as well as being quite versatile in how they can assist humans.

However, we have also discussed the limits of these systems, and how what appears to look like intelligence is sometimes an illusion made possible by the vast amounts of data ingested by these systems:

- First, none of these systems is nowhere near being generic. While they are somewhat more versatile than their predecessors, they remain limited to a narrow range of tasks: the fact that we have 3 categories of AIs discussed in this paper is proof enough of the lack of general purpose.

- While assistant AIs appear to have reasoning and common sense abilities, they fail on a very regular basis to solve very simple problems, which shows that this illusion of reasoning and common sense is mainly due to brute force learning. The same could be said of gaming AIs that play moves that are difficult to explain for a humans, but sometimes fail to see that the game is over and they have lost in situations where it would be impossible to miss for a human.

- The idea of creativity is very subjective and difficult to evaluate, even for humans. We could argue that most of what appears to be creative with these systems is merely the statistical result of very long adversarial learning sessions with a bit of randomness here and there. And what to say about the hallucinations that LLMs systems have when they make up things because they have to answer something ? Is it creativity, or a proof of stupidity ?

- Finally, we have seen that the adaptability and genericity of these systems is severely hindered by the neural network architectures that power them: They are limited in terms of how they can interface with the world, and their ability to evolve is constrained by an architecture that needs to be retrained if it is modified.

- We have also shown that this same architecture currently prevents the explosion of intelligence, the *singularity* that some scientists from fields outside of artificial intelligence appear to fear.

Some scientists even go as far as saying that any illusion of emergent abilities in the latest versions of these systems (that is LLMs) can easily be debunked with the right statistical tools and enough time (Schaeffer, Miranda, & Koyejo, 2023).

In short, we can say that despite undeniable and very impressive progresses, current AIs systems are very far from achieving AGI. We could even argue that they are not really in the right direction because of the limitations imposed by neural network based architectures in terms of volumes of data required to train them, interfacing constraints, lack of symbolic learning abilities and how these models are trained.

4.2 Challenges, Dangers and Opportunities

Being far from AGI does not mean that these systems and their future iterations won't have a major impact on our society. While they are not AGI, these systems have proven to be better and faster than humans in several key tasks: Deep fakes are already a major source of both amusement and disinformation at the same time. Large language models, which are the current trend, will most likely destroy a large number of assistant and white collar creative jobs, as well as call center jobs (such as customer services) in the coming few months (Zarifhonarvar, 2023). Another example of potential disruption that these AIs may cause was highlighted during the recent writers and actors strike in Hollywood (Dalton & {The Associated Press}, 2023) with -among other things- the former fearing replacement by ChatGPT to write the scenarios of series to come, and the latter fearing the use of Generative AIs to use their younger image and voice without control for an infinite amount of time.

On the other hand, they will also make knowledge easily available and summarized to anyone with an internet connection. These AIs will also make basic programming languages a lot easier and less time consuming. Likewise, they will be invaluable for translation and sumarization tasks, as well as for bibliographic works. While creative jobs such as programmers, writers, artists and assistants may see them as rivals and dangers, these AIs may well prove to be valuable helpers rather than relentless and cheap labor replacement. There may also be job creations dedicated to cutting complex problems into simple ones that these models can understand and fit within their limited input interfaces to achieve the best possible results.

However, due to the way they are trained, their nature as black box neural networks (whose answers are difficult to explain), and their tendency to have hallucination, there will be real challenges around assessing the reliability of these models, especially large language models (Liu et al., 2023). Indeed, the correct use of these tools may prove difficult for users unaware of such issues, and that these systems are not the perfect oracles we often depict them to be. They tell you (or write the piece of code) that they think you want to see, rather than a correct answer. This may lead to a large number of legal issues regarding who should be responsible when one of these system will fail: The designers of the AI ? The people who chose the data it was trained with ? The scientists tasked with the impossible mission of assessing the system reliability and putting an overlay of safeguards ? The company or final user of the trained system ?

Furthermore, when it comes to both large language models and artistic AIs, since they produce their answer (image, video or text) based on large datasets they have learned, copyright questions may arise due to some answers being full, partial or piece by piece copy-past of existing texts, codes and images that might be copyrighted. This may be particularly challenging with most companies keeping their training data and processes secretive, and the data being impossible to guess once these algorithms are trained.

It is also easy to see how modern warfare could use systems similar to the gaming AIs discussed in this paper. It is hard to tell if this is a good or a bad thing, but countries without them will be at a severe disadvantage. Furthermore, we should also be wary of the limits discussed for these systems and what the consequences of the so-called *hallucinations* and other strange behaviors out of the regular scope would mean in the context of AIs being

generalized in health systems, or for AI facing one another on financial markets. To this day, it is impossible to tell.

Finally, since these systems can replace humans for many basics tasks such as drawing, programming, translation, computation, summarizing of documents, etc; we may wonder what could happen to a society where nobody practices basic skills anymore since AIs can do it better and faster. Indeed, while it may seem convenient, many of these basics skills are also a necessary basis (hence their name) to learn more complex skills and tasks that AI cannot (or not yet) do. As such, and given the lack of real creativity of these systems, there might also be a technological stagnation risk when they will have reached their limits as we have discussed them in this paper, and when there might be no human skilled enough left to feed them with higher level knowledge or skills due to an erosion of basic skills knowledge in the scientific population.

We may therefore conclude that the danger with currently emerging AI systems does not lie in their supposed intelligence, but on their short-comings as well as wrongly using them: It is difficult to predict the consequence of such systems -deployed on a global scale- reaching their limits and being used in ways that we either did not expect, or that -by design- they simply cannot handle. There is also a major issue with the reliability of the results they produce: While they do fine in most cases, how do we detect the cases where they don't ? Furthermore, there are legal aspects both in terms of responsibility, but also in terms of copyrights and image rights due to their answers being produced from existing content they absorbed during their training which will need to be considered. Finally, we may come back to the question of delegating too much to these systems, this time not because of the reliability risk, or the risk of job destruction, or not even the environmental cost of these systems (which we have not discussed in this paper), but because of the unknown consequences of delegating to many basic tasks to machines and the risk of mass stagnation that may occur.

References

Amari, S. (1967). A Theory of Adaptive Pattern Classifiers. *IEEE Transactions on Electronic Computers*, *EC-16*(3), 299–307.

Armstrong, S., Bostrom, N., & Shulman, C. (2016). Racing to the precipice: a model of artificial intelligence development. *AI & SOCIETY*, *31(2)*.

Berg, M. (2023). 'Four Battlegrounds' shaping the U.S. and China's AI race. *Politico*.

Borji, A. (2023). A Categorical Archive of ChatGPT Failures. *arXiv*.

Bostrom, N. (2017). Strategic Implications of Openness in AI Development. *Global Policy*, *8*, 135–148.

Bowman, S. R. (2023). Eight Things to Know about Large Language Models. *arXiv*.

Brown, T. B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T., Child, R., Ramesh, A., Ziegler, D. M., Wu, J., Winter, C., Hesse, C., Chen, M., Sigler, E., Litwin, M., Gray, S., Chess, B., Clark, J., Berner, C., McCandlish,

S., Radford, A., Sutskever, I., & Amodei, D. (2020). Language Models are Few-Shot Learners. *CoRR, abs/2005.14165*.

Bubeck, S., Chandrasekaran, V., Eldan, R., Gehrke, J., Horvitz, E., Kamar, E., Lee, P., Lee, Y. T., Li, Y., Lundberg, S., Nori, H., Palangi, H., Ribeiro, M. T., & Zhang, Y. (2023). Sparks of Artificial General Intelligence: Early experiments with GPT-4. *arXiv*.

Castro, D., McLaughlin, M., & Chivot, E. (2019). Who Is Winning the AI Race: China, the EU or the United States?. Tech. rep., Center for Data Innovation.

Cave, S., & Dihal, K. (2019). Hopes and fears for intelligent machines in fiction and reality. *Nature Machine Intelligence*, *1*, 74–78.

Dalton, A., & {The Associated Press} (2023). Writers strike: Why A.I. is such a hot-button issue in Hollywood's labor battle with SAG-AFTRA. *Fortune*.

Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2019). BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pp. 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., & Bengio, Y. (2014). Generative Adversarial Nets. In Ghahramani, Z., Welling, M., Cortes, C., Lawrence, N., & Weinberger, K. (Eds.), *Advances in Neural Information Processing Systems*, Vol. 27. Curran Associates, Inc.

Goodfellow, I. J., Shlens, J., & Szegedy, C. (2015). Explaining and Harnessing Adversarial Examples. In *ICLR (Poster)*.

Han, T. A., Pereira, L. M., Santos, F. C., & Lenaerts, T. (2020). To regulate or not: A social dynamics analysis of an idealised AI race. *J. Artif. Intell. Res.*, *69*, 881–921.

Heaven, W. D. (2020). DeepMind's protein-folding AI has solved a 50-year-old grand challenge of biology. Tech. rep., MIT Technology Review.

Hinton, G. E., & Zemel, R. (1993). Autoencoders, Minimum Description Length and Helmholtz Free Energy. In Cowan, J., Tesauro, G., & Alspector, J. (Eds.), *Advances in Neural Information Processing Systems*, Vol. 6. Morgan-Kaufmann.

IBM (2008). Deep Blue – Overview. *IBM Research*.

Kingma, D. P., & Welling, M. (2014). Auto-Encoding Variational Bayes. In *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*.

Lake, B. M., Ullman, T. D., Tenenbaum, J. B., & Gershman, S. J. (2017). Building Machines That Learn and Think Like People. *Behavioral and Brain Sciences*, *40*.

LeCun, Y., Boser, B., Denker, J. S., Henderson, D., Howard, R. E., Hubbard, W., & Jackel, L. D. (1989). Backpropagation applied to handwritten zip code recognition. *Neural Computation*, *1*(4), 541–551.

LeCun, Y. (2018). The Power and Limits of Deep Learning. *Research-Technology Management*, *61*(6), 22–27.

Lee, K.-F. (2018). *AI superpowers: China, Silicon Valley, and the new world order*. Harper Business.

Legg, S., & Hutter, M. (2007). A Collection of Definitions of Intelligence. In *Proceedings of the 2007 Conference on Advances in Artificial General Intelligence: Concepts, Architectures and Algorithms: Proceedings of the AGI Workshop 2006*, p. 17–24, NLD. IOS Press.

Liu, R., Pang, Z., Meng, Z., Wang, W., Yu, Y., & Lu, T. (2022). On Efficient Reinforcement Learning for Full-length Game of StarCraft II. *J. Artif. Intell. Res.*, *75*, 213–260.

Liu, Y., Yao, Y., Ton, J.-F., Zhang, X., Guo, R., Cheng, H., Klochkov, Y., Taufiq, M. F., & Li, H. (2023). Trustworthy LLMs: a Survey and Guideline for Evaluating Large Language Models' Alignment. *arXiv*.

Maharaj, S., Polson, N., & Turk, A. (2022). Chess AI: Competing Paradigms for Machine Intelligence. *Entropy*, *24*(4).

Maslej, N., Fattorini, L., Brynjolfsson, E., Etchemendy, J., Ligett, K., Lyons, T., Manyika, J., Ngo, H., Niebles, J. C., Parli, V., Shoham, Y., Wald, R., Clark, J., & Perrault, R. (2023). The AI Index 2023 Annual Report. Tech. rep., Institute for Human-Centered AI, Stanford University.

MetaAI (2023). Introducing LLaMA: A foundational, 65-billion-parameter large language model. *arXiv*.

Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., & Dean, J. (2013). Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pp. 3111–3119.

Minsky, M., & Papert, S. (1969). *Perceptrons: An Introduction to Computational Geometry*. MIT Press, Cambridge, MA, USA.

Naughton, J. (2023). As AI weaponry enters the arms race, America is feeling very, very afraid. *The Guardian*.

OpenAI (2023). GPT-4 Technical Report. *arXiv*.

Pearl, J. (1982). The Solution for the Branching Factor of the Alpha-Beta Pruning Algorithm and Its Optimality. *Commun. ACM*, *25*(8), 559–564.

PwC (2017). Sizing the prize: What's the real value of ai for your business and how can you capitalise?. Tech. rep., PwC London.

Reich, A. (2023). 1/3rd of scientists fear ai decisions could spark nuclear-level disaster - report. *The Jerusalem Post*.

Rumelhart, D. E., Hinton, G. E., & Williams, R. J. (1986). Learning representations by back-propagating errors. *nature*, *323*(6088), 533–536.

Schaeffer, J. (2014). *One Jump Ahead: Challenging Human Supremacy in Checkers*. Springer.

Schaeffer, R., Miranda, B., & Koyejo, S. (2023). Are Emergent Abilities of Large Language Models a Mirage?. *arXiv*.

Schartner, M. M., & Timmermann, C. (2020). Neural network models for DMT-induced visual hallucinations. *Neuroscience of Consciousness*, *2020*(1).

Shanahan, M. (2015). *Front Matter*, pp. i–vi. The MIT Press.

Silver, D., Huang, A., Maddison, C. J., Guez, A., Sifre, L., van den Driessche, G., Schrittwieser, J., Antonoglou, I., Panneershelvam, V., Lanctot, M., Dieleman, S., Grewe, D., Nham, J., Kalchbrenner, N., Sutskever, I., Lillicrap, T., Leach, M., Kavukcuoglu, K., Graepel, T., & Hassabis, D. (2016). Mastering the game of Go with deep neural networks and tree search. *Nature*, *529*(7587), 484–489.

Silver, D., Hubert, T., Schrittwieser, J., Antonoglou, I., Lai, M., Guez, A., Lanctot, M., Sifre, L., Kumaran, D., Graepel, T., Lillicrap, T., Simonyan, K., & Hassabis, D. (2018). A general reinforcement learning algorithm that masters chess, shogi, and Go through self-play. *Science*, *362*(6419), 1140–1144.

Silver, D., Schrittwieser, J., Simonyan, K., Antonoglou, I., Huang, A., Guez, A., Hubert, T., Baker, L., Lai, M., Bolton, A., Chen, Y., Lillicrap, T., Hui, F., Sifre, L., van den Driessche, G., Graepel, T., & Hassabis, D. (2017). Mastering the game of Go without human knowledge. *Nature*, *550*, 354–359.

Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., & Rabinovich, A. (2015). Going deeper with convolutions. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1–9.

Szegedy, C., Zaremba, W., Sutskever, I., Bruna, J., Erhan, D., Goodfellow, I., & Fergus, R. (2014). Intriguing properties of neural networks. In *ICLR (Poster)*.

Thoppilan, R., Freitas, D. D., Hall, J., Shazeer, N., Kulshreshtha, A., Cheng, H., Jin, A., Bos, T., Baker, L., Du, Y., Li, Y., Lee, H., Zheng, H. S., Ghafouri, A., Menegali, M., Huang, Y., Krikun, M., Lepikhin, D., Qin, J., Chen, D., Xu, Y., Chen, Z., Roberts, A., Bosma, M., Zhou, Y., Chang, C., Krivokon, I., Rusch, W., Pickett, M., Meier-Hellstern, K. S., Morris, M. R., Doshi, T., Santos, R. D., Duke, T., Soraker, J., Zevenbergen, B., Prabhakaran, V., Diaz, M., Hutchinson, B., Olson, K., Molina, A., Hoffman-John, E., Lee, J., Aroyo, L., Rajakumar, R., Butryna, A., Lamm, M., Kuzmina, V., Fenton, J., Cohen, A., Bernstein, R., Kurzweil, R., y Arcas, B. A., Cui, C., Croak, M., Chi, E. H. , & Le, Q. (2022). LaMDA: Language Models for Dialog Applications. *CoRR*, *abs/2201.08239*.

Vinge, V. (1993). The coming technological singularity: How to survive in the post-human era. In NASA (Ed.), *Vision-21: Interdisciplinary Science and Engineering in the Era of Cyberspace*, Vol. 10129 of *NASA Conference Publication*, pp. 11–22, Cleveland, OH. NASA Lewis Research Center.

Waibel, A., Hanazawa, T., Hinton, G., Shikano, K., & Lang, K. (1989). Phoneme recognition using time-delay neural networks. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, *37*(3), 328–339.

Wei, J., Tay, Y., Bommasani, R., Raffel, C., Zoph, B., Borgeaud, S., Yogatama, D., Bosma, M., Zhou, D., Metzler, D., Chi, E. H., Hashimoto, T., Vinyals, O., Liang, P., Dean, J., & Fedus, W. (2022a). Emergent Abilities of Large Language Models. *arXiv*.

Wei, J., Wang, X., Schuurmans, D., Bosma, M., Xia, F., Chi, E., Le, Q. V., Zhou, D., et al. (2022b). Chain-of-thought prompting elicits reasoning in large language models. *Advances in Neural Information Processing Systems*, *35*, 24824–24837.

Xaio, H. (2023). Auto-GPT Unmasked: The Hype and Hard Truths of Its Production Pitfalls. *Jina AI*.

Xiang, C. (2023). Microsoft Now Claims GPT-4 Shows 'Sparks' of General Intelligence. *Vice*.

Zarifhonarvar, A. (2023). Economics of chatgpt: A labor market view on the occupational impact of artificial intelligence. *Available at SSRN 4350925*.

Zhou, D., Schärli, N., Hou, L., Wei, J., Scales, N., Wang, X., Schuurmans, D., Cui, C., Bousquet, O., Le, Q. V., & Chi, E. H. (2023). Least-to-Most Prompting Enables Complex Reasoning in Large Language Models. In *The Eleventh International Conference on Learning Representations*.