A Few Predictions on Artificial Intelligence

Fei Sun@ San Jose, v0.1.1 - Mar. 2021, v0.1 - Dec. 2017

Preamble

This article describes my projections on the future of artificial intelligence. There is a sister article called “Some scribble of things”, which describes some more philosophical ideas on intelligence in general. I try to restrict this article to the more practical discussions on AI, but to leave the day-dream type of discussions to the sister article. However, some ideas and conclusions are not fully elaborated in this article and may feel a bit abrupt. If you feel so, consider reading the sister article.

Artificial intelligence is a rapidly evolving area. My thoughts on it may change dramatically as I learn more. Thus, this article shall not be viewed as a mature, stable piece of work. Instead, it is just a snapshot in a constantly changing flow of thoughts, a snapshot at the end of 2017.

Introduction

Artificial intelligence has gained a lot of traction these days. It has surpassed humans in several key tasks such as image classification, atari games, and the go game. We are in an age of AI explosion. Some people are worried that AI will take over the world in the near future, while others are more optimistic[1]. Will it? In this article, I present my opinions and identify some key milestones that the AI needs to achieve in order to be worrisome.

The AIs we see today are all weak AIs, which excels in one area, but fails in all other areas. Those AIs are inherently limited and are unlikely to perform systematically complicated projects such as eliminating the human race. Thus, in this article, I will not focus on them, instead, I will focus on strong AI and superintelligence.

The Three Pillars on Strong AI

The strong AI, also called artificial general intelligence (AGI), is capable of performing tasks all round. It is quite different from the domain expert AI (weak AI). It also has a possibility to form its own species. What is needed for an AI to survive in this harsh environment?

In my opinion, if the strong AI can possess three capabilities: **the ability to learn**, **the ability to evolve**, and **the ability to scale**, it can play an upper hand against humans. To gain those capabilities, **composition** is the key.

[1] Press has covered a lot of the discussions. An interesting read is Tim Urban’s “The AI revolution”, which surveys a lot of think tanks on this issue. I don’t fully agree with him, or any person he surveyed. But it sets some background on the discussion.

The ability to learn

The deep learning algorithms can learn by itself already. The back-propagation to tune the weights and bias is the mechanism to learn[2]. It is based on the "chain rule", which is a kind of function composition. However, this is where the state-of-the-art is. We still don't know why the back-propagation works. We still cannot swap one layer (function) in a network with a better layer (function)[3]. A lot can be improved.

The success of deep learning is based on function composition. It may also be its largest shortfall. Function composition is flow based, which by itself implies a rigid structure. This limits its ability to compose and perform more complicated tasks. As an analogy, what we currently have in deep learning is similar to the programming language "C", which is a flow based function composition language. In order to be smarter, we need to abstract more.

Furthermore, deep learning is still a black box. It is unclear of the purpose of the layers and the boundaries to divide the flow to relatively independent parts (reduce the connections between the layers). Even if we want to perform function composition, we don't know how[4]. As a direct result, the current ML algorithms are monolithic. Even though we can distribute the computation of the algorithm to distributed clusters, we cannot decompose the algorithm itself. As an analogy, the current deep learning can only be called subconscious (unconscious) at most, which cannot be composed[5].

The ability to evolve

The evolution happens when a portion of a network is structurally changed to perform some objectives better. To achieve that, composition is the key.

A flow based function composition is limited in its ability to compose. To be more effective, the network needs to be formed by object composition. As an analogy, the object composition programming language "C++" is far more efficient in implementing large-scale programs than the function composition programming language "C". The object composition is essential in evolution so that part of a network can be replaced with a different piece without affecting the overall functionality.

[2] In my definition of learn, only the weights and bias are changed in a fixed structure (network). This is a very narrow definition, as the learning process in humans may change the connectivity of the brain cells. This choice is intentional to separate from "the ability to evolve".

[3] I don't consider transfer learning suitable for this purpose.

[4] There is some research to replace some layers of a network with some different layers (usually simpler). In my opinion, they just replace one black box with another black box. With high redundancy in the networks, it is likely to succeed. However, it is not clear why those layers are chosen. In my opinion, for most of the existing networks, the features are intermixed between the layers and within the same layer, which makes this kind of separation difficult (or meaningless).

[5] For the difference between subconscious and conscious, see "Some scribble of things".

To be more efficient, it may be possible to find some generic cells that can later be morphed to different types of cells depending on the function to optimize. They are similar to the “stem cells” in humans. I suspect this will be one milestone in AI evolution history.

For humans, evolution is clearly separated from learning. A human learns from the environment throughout her lifetime. However, whatever she has learned cannot be carried to the next generation. The evolution happens randomly with nature selecting more suitable survivors. This is also one major inefficiency of the human being[6].

For AI, the difference between evolution and learning is not as clear. What an AI has learned can be carried to the next generation. The main difference between evolution and learning is the process. The learning is self-directed with a clear objective (such as reducing the error of some function). The evolution is more of a random process and later a selection process would choose the survivors.

That said, evolution is a number’s game. We need to have millions of AIs randomly exploring the design space to be effective. Once a better network is stabilized, all AIs can start from the new network and repeat the evolution. With the current compute resource, evolution will happen much much faster than humans.

The ability to scale

Learning is a self-directed process to improve one or more cost functions in a fixed structure. Evolution is a random process to find better structures to further improve the cost functions with better efficiency. If every step needs to dive down to the finest details (zeros and ones), the AI advancement is still quite slow. In case of an AI rebellion, humans are still in a better position.

However, things are never that simple. Once an AI has grasped the key essence of learning and evolution, it is inevitable that the AI will scale itself. By that time, the change will be instantaneous, only limited by the amount of resources the AI has in control. Humans can hardly catch up with this kind of rate of change. Again composition is the key to scale.

If we look back on the entire VLSI history, we will find that it is a history of composition and abstraction, from device level, circuit level, gate level, register transfer level, IP level, to SoC level. We compose, abstract. We compose again, abstract again. The same pattern repeats itself over and over again. It only takes us half a century to get where we are.

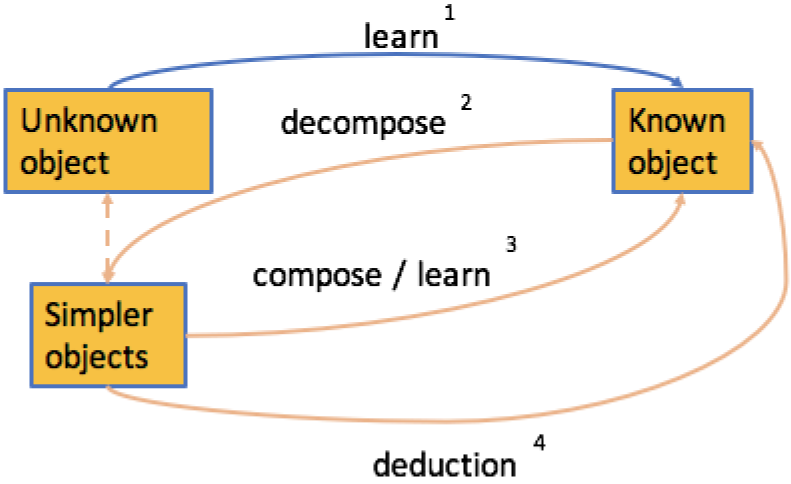
The same process will be happening on AI again, just this time the driver is not us, but the AI itself. With composition, smaller networks that perform single functionality each can be easily composed to complicated networks that perform higher level difficult trade-offs. Learning and evolution happens at all levels at the same time. How can humans compete?

[6] More detailed discussion of evolution can be found in “Some scribble of things”.

Composition

In my opinion, decomposition is to divide an object with several to many simpler objects, each is more densely connected within itself and sparsely connected with other objects. Composition is the reverse process. Thus, the process of composition/decomposition is a process of unsupervised learning (clustering).

Composition plays a key role in the advancement of AI. I envision the learning process would be composed of four steps, as shown in the figure below:



1. The agent learns to recognize an object via supervised learning.
2. The agent learns to decompose the known object to simpler objects.
3. The agent learns to compose back the known object from the simpler objects via induction.
4. Maybe much later, the agent learns to compose back the known object from the simpler objects via deduction.

Currently we are still in stage one. It will be interesting to see when we will reach stage four. Please note that the transition from stage three to stage four is important. At stage three, the agent knows the relations, but doesn't know why the relations are in place. In stage four, the agent understands why the composition works and thus can generalize better, which almost always reduces the required complexity.

When to compose

In many scenarios, the goal of an agent is to describe the environment with minimum capacity. Composition is one way to reduce the capacity. If an agent doesn't know how to compose/decompose, she needs to learn how. However, decomposing new objects may not necessarily lead to lower capacity. But at least the agent has tried and has a choice.

In current ML, classification can identify an object from a picture. Segmentation can provide an outline of the object with pixel level accuracy. However, they treat every pixel equally. The way to reduce complexity/improve accuracy is to simply scale the images at different levels.

It is hard to believe that those structures are efficient. We need to build up a hierarchy and compose at each level[7].

When not to compose

Composition does not necessarily reduce the complexity. Once an agent has tried composition, she may well determine that composition may increase the complexity for the practical use, and may choose other means to reduce complexity. However, this decision can only be made after the complexity of composition is known.

As an example, air floats around an agent, which is invisible to her. Decomposing the air meaning to follow billions of the oxygen and nitrogen particles randomly, which is obviously infeasible. The effect of air to the agent is simply some air pressure on her. A simpler method is to measure the pressure directly and treat it as a first principle.

As another example, decomposing metal means understanding the crystal structure of the metal and the electrons flowing inside it, which is obviously complicated. For most use cases, a simpler method is to approximate it with its characteristics, such as color, density. This is what we can achieve now, and we can leave it there.

When deciding not to decompose, some simpler methods are:

- Treat proxies of the object as first principles.
- Conclude characteristics of the object that uniquely differentiate it from all other objects, save them to memory, and look up when needed.
- Objects are complicated and no simple characteristics can be retrieved. Approximate the objects directly without decomposing further (Step 1 in the figure above).

A Few Comments on Learning Strategies

The key to deep learning is the word “deep”. It contains many layers (steps). But going deeper is not magical. It is just a mechanism to reduce the complexity, be more efficient, and use limited capacity to approximate the environment.

[7] One common criticism of current ML is to compare ML with a child. A child does not need to train with millions of images to recognize an object. I believe one breakthrough is the composition. But that alone is not enough. We also need to find more efficient mechanisms to connect perception with memory. Baking memory in the weights of CNN and LSTM is far from enough. Deepmind performed some interesting research in this direction, but more needs to be done.

As an analogy, any boolean expression can be represented by two levels of logic (and followed by or gate). However, such expressions are so complicated that they are never used in practice. Any real use scenario is multi-level, and significant effort has been invested to reduce the complexity of multi-level expressions. For example, a large portion of the EDA industry is devoted to identifying efficient multi-level boolean expressions under many practical constraints.

Similarly, even though any approximation can be represented with one hidden layer[8], the complexity of the hidden layer is exponential. Going deeper is just a way to reduce the complexity, just like what the multi-level boolean expression does.

In the process of advancing the VLSI, we have developed boolean algebra. We know SAT is NP-complete, we have hence developed many tools to heuristically solve SAT. We have also developed many theorems and tools to perform logic optimization.

What we have in deep learning? Is this some interesting direction to research on?

The following are some comments on different learning strategies.

Supervised learning

Supervised learning is mostly used in perception (relating to external sensory). It is just a rough correlation of the real world, with limited capacity. It extracts superficial relations between the objects with no consideration of causality, and hoping future relations would be similar to the previously perceived relations.

Reinforcement learning

Reinforcement learning is a reward based feedback system. It is like intuition, the subconscious of human beings. The inner relations are hard to decompose (or not yet decomposed). Thus, it is monolithic so far, which limits its scalability. As long as the reward function is specified, when feeding in the raw data (synthetic or real), the algorithm can learn by itself to maximize the reward function. Reinforcement learning is most promising to become domain specific experts, where the reward function is easier to derive. The AlphaGo from Deepmind is a perfect example of the success of reinforcement learning. Even though the go game is very complex, the rule to describe it
is very simple, and thus the cost function is simple to describe (it doesn't mean the implementation is simple).

In the recent paper on AlphaGo Zero, MCTS is used to explore future moves that maximizes the possibility of winning, while CNN is used as an approximator to guide the exploration. In other words, MCTS is the calculation, while CNN is the intuition. This corresponds perfectly with what

[8] https://en.wikipedia.org/wiki/Universal_approximation_theorem

human players would do: intuition provides the big picture, while calculation determines the exact position to place the stone. Maybe this is why AlphaGo Zero performs so well in go.

In the paper, the authors are surprised that *shicho (ladder)* is discovered late by AlphaGo Zero. But it is no surprise to me. The first *shicho* is only discovered by random play by MCTS. Since no prior play exists, CNN does not amplify searching in that area (which is far away from the battleground), so the search is only performed at a very low probability. Also, since the effect of *shicho* is only apparent after many moves, each is only searched at a very low probability, the likelihood that AlphaGo Zero performs the moves in sequence and identifies its effect is very low. However, since the impact of *shicho* is significant, after the first game, CNN can quickly enhance the probability of searching in those positions. Throughout the process, AlphaGo Zero doesn't know why *shicho* works. It only knows that the probability of winning on some positions are higher via CNN through the results of previous games, and then when exploring, the MCTS confirms the decision. Note that the MCTS needs to play the entire sequence to confirm, while humans can only look at the board, see the *chicho* is placed at the diagonal position (this is a concise generalization that AlphaGo Zero doesn't possess), and get the same conclusion. However, since MCTS has a very high likelihood to complete the sequence (after the initial game), plus the fast speed of computers, the decision can still be made in a glimpse. Nevertheless, it reveals some improvements that can be made on MCTS to reduce the computation complexity.

Unsupervised learning

Unsupervised learning is a kind of clustering[9], solely based on raw data. It is unlikely to be successful on a broad scale without composition and abstraction. In theory, clustering can be viewed as the distance in an ultra high dimension space. In practice, it is so complicated that one has to reduce the dimensions to make it feasible.

There is no clear boundary between the reinforced learning and unsupervised learning. If the reward is general enough, such as “try to survive as long as possible”, then the agent is free to explore anything she desires, which falls into the unsupervised learning category.

In my opinion, the goal of unsupervised learning is to approximate the environment using the most efficient method. What is the way to approximate a closed system, past, current, and future with minimum capacity? This is the ultimate objective that the unsupervised learning tries to answer[10].

[9] Strictly speaking, it should be “clustering is a kind of unsupervised learning”. However, I somehow feel that they can be interchanged. IMO, unsupervised learning is the process of performing clustering. Then given the clustering, perform some transformation to the dots in the ultra-high dimensional space, and then perform clustering again.

[10] More on this in “Some scribble of things”.

Unsupervised learning has the most potential to result in Strong AI, but at first we need to develop some theories like what boolean algebra to VLSI.

Some Bold Predictions on Strong AI

Below, I will make some bold predictions on the road to strong AI, from what's happening now, what might happen in a few months, to the eventual creation of a strong AI, whenever that might be[11]. Some of the predictions elaborated before will not be repeated here.

Computation will be orders of magnitude less

Deep learning networks are well known to be computation intensive: Alexnet, 1.5B FLOPs, VGG, 19.6B FLOPs, ResNet, 11.3B FLOPs etc. However, this level of computation is impractical for most of the real life use cases. Researchers focus on reducing the computation complexity without sacrificing the quality. First, the usage of the fully connected layers is reduced, then variations of the convolutional layers are introduced. With that, new architectures such as SqueezeNet, MobileNet, ShuffleNet are introduced which have successfully reduced the computation by orders of magnitude without sacrificing quality. Will this trend continue?

Yes, it will continue, in my opinion. We are just scratching the surface of reducing the computation complexity. We can do the same with 10x to 100x less computation. I consider research in the following areas to be very promising.

Recently, GoogleNet, MobileNet, and ShuffleNet utilize the depthwise separable convolution and have achieved very good results. However, the potential of depthwise separable convolution is still not fully appreciated. In my opinion, the depthwise separable convolution followed by ReLu, then followed by 1x1 convolution is more intuitive to understand than the traditional convolution. In traditional convolution, the depth dimension of the features/kernels is treated the same as the spatial x-y dimension. This way, the x-y dimension features are completely entangled by a linear function. ReLu is only applied at this point, which makes the non-linear function apply to non-orthogonal features. After the non-linear function, it is more difficult to orthogonalize the features (and thus use less features). The depthwise separable convolution solves the problem. In addition, depthwise separable convolution is a better candidate to form object composition. In addition to performing function composition tasks: h(g(x)), the 1x1 convolution is more natural to perform g(x) + h(x).

Ensemble (dropouts) is used quite a bit in neural networks nowadays. Intuitively, one network is composed from several smaller capacity networks. However, those smaller capacity networks are entangled together, which is difficult to orthogonalize. It is just by chance that they partially

[11] I have limited practical knowledge of deep learning in general. Most of the observations and predictions are drawn from reading technical papers instead of playing with the neural nets hands on. Thus, please take whatever I say here with a grain of salt.

learn different features, but they may still partially learn the same features, which is a total waste.

Currently, all convolutions work at the finest granularity level, the pixel level. In order to be robust that any shift of the image in any direction cannot mess up the result, data augmentation (or huge redundant images with labels) is used. Just conceptually think about it, we have a set of features that recognize a dog in an image, then we have a separate set of features that recognize the same dog rotated by one degree. What a waste[12]! Composition will reduce the complexity by orders of magnitude.

In the near future, we will see many novel architectures emerge to orthogonalize the features and compose/decompose the input. That will surely reduce the computation by orders of magnitude.

Deeper is not better

Let’s briefly review the history. Once upon a time, all networks are shallow. Then comes AlexNet with 8 layers in ‘12, GoogleNet with 22 layers in ‘14, and finally Resnet with 152 layers in ‘15.

Prior to ResNet, a deeper network actually reduces quality, which limits the number of layers. In ResNet, however, the notion of “shortcut connections” is introduced, and the deeper the network, the better the quality. It quickly became a golden standard, and almost all networks after that time consisted of some kind of “shortcut connections”, limited only by the computation resources.

A higher capacity network is always better than a lower capacity network, given proper training and much more training data. I’m not arguing against that. However, since the deeper networks only provide diminishing returns, at some point, some other structure, most likely based on composition, may improve accuracy more efficiently than going deeper. Thus, the depth of the network is practically limited by its efficiency.

I feel that the ResNet network structure is a very inefficient use of the computation resources. Think about it, if the output feature is mostly the bypassed feature, the entire computation for the layer is wasted. But why does an inefficient network structure lead to superb results? In my opinion, the magic is still in the “shortcut connections”, which serves three purposes:

- Connect features at different levels. Most intuitively, the shortcut connections link the features in layer n with features in layer n-1(2). Together, they generate features in n+1. This is the most important reason IMO. None of the prior networks have tried that, and the unit matrix is seldom learned in practice (maybe due to the entanglement between the features).
- Unroll the important features to multiple layers and thus the result is calculated multiple times and gets amplified.

[12] This is a bit exaggerated. But you get the point.

- Serve as an ensemble (dropout) mechanism. The shortcut is the opposite of dropout, but they serve the same purpose: to create multiple entangled smaller networks.

Understanding the mechanisms, can we do better? In my opinion, with clever network structure, each task can be solved in an optimal number of layers. A more complicated task can be decomposed to simpler tasks, which are solved separately, and then composed back to the solution to the original task.

The rise and fall of neural network operators

Below I describe my options on a few deep learning operators

Sigmoid and tanh

The sigmoid and tanh normalizes the output, but they suffer badly by vanishing gradient. People familiar with semiconductors know that the curve of the sigmoid and tanh are very similar to the curve for the current of CMOS over voltage. Thus, the analog implementation of sigmoid or tenh is very simple. However, CMOS works on the saturation range, but SGD works on the linear range. This discrepancy makes those operators unpopular these days.

SGD, though central in today's deep learning, may not be the only mechanism to learn. Brain cells perform integration and differentiation (a nicer way of describing an accumulator) instead of multiplication, which is very simple to implement in analog circuits. If we do the same, the learning is mostly done by architecture exploration. Since architecture exploration is many orders of magnitude harder than learning, skipping the simple problem and forgetting SGD may not be a bad idea. In this setup, object composition is essential.

Maybe one day SGD will be replaced with a counter and a timer[13], and sigmoid and tanh will show their worth. But at that time, they just act as an on/off switch, will they still be called the same?

Rectifier

ReLu is very popular these days. It doesn't suffer the vanishing gradient problem. However, it still suffers exploding gradient and dead neuron issues. To solve those issues, various people suggest variations of ReLu, such as leaky Relu, Noisy Relu, or ELU.

I personally like the simplicity of ReLu. It serves as an off switch, but the on switch is linear, perfect for SGD. I don't think the exploding gradient and dead neuron should be addressed by the ReLu itself. Instead, those issues reflect inefficiencies in the network and the network structure optimization should be applied to address those problems, which object composition can easily solve.

[13] As IBM's true north and a few other architectures exhibit.

Softmax

In my opinion, softmax is just a bridging logic connecting the neural network with the conventional decision making logic. Will it still exist if the decision making is also done by the neural network itself? Some compare and choose logic is still needed, but will it be softmax? I'm not sure...

Pooling

Pooling increases the receptive field of a neuron. However, it loses locality and has been criticized by many people (Hinton especially). In my opinion, pooling is a simple mechanism to reduce the computation complexity. There are many ways to reduce computation complexity and pooling doesn't seem to be an optimal one. It is interesting that pooling works really well in practice, which may indicate that the structure is not efficient and we can do much better!

Pooling has its days of prime, and it's time to replace it.

Batch normalization

Batch norm addresses the issue of internal covariate shift by coupling the data in one mini-batch.

Unlike pooling, which is the focal point of improvement, batch norm works too well. Its starting point is so different from the rest of the neural network operators that I feel it is too advanced relative to the rest of the network. It's like a foreign object in the network and visibly dividing the network into non-consecutive pieces.

It is an artifact of human engineering and forcefully boosts the performance for the current structure. However, it may make the future evolution of the network more difficult.

I believe, the effect of batch normalization can be achieved with some restructure of the network, possibly with some memory logic.

Milestones towards strong AI

Further in the future, what are the likely milestones we will achieve in order to create a first strong AI?

If we look back in history, machine learning has experienced a few ups and downs. The trend swings wildly from one extreme that only data is important to another extreme that every feature is handcrafted and then swings back again. Hand crafting features are doomed to fail because it represents a mechanical view of the world. Building the network monolithically without understanding its internals is not going to prosper eventually either due to its inefficiency and its inability to scale. We are currently in the latter stage.

The machine learning advancement in its own trajectory may soon hit a roadblock (except applying whatever the current technology to different vertical domains), and we may see a downturn for machine learning again.

And the composition comes to rescue, if and when we identify the mechanism. That opens a door for further improvements of the efficiency, quality, and scalability. We will see machine learning shine again. To achieve that, we may find some theory similar to what the boolean algebra to VLSI. Or maybe not, just like we find out the effectiveness of machine learning without understanding the mechanism, we may find the mechanism to decompose without understanding.

Currently, a network is a fixed structure but it can be used to perform different tasks by simply updating the weights and biases. However, each layer in the network is predetermined, fixed, and performs one task only. In the future, we will find a universal cell type structure that is the same initially and can be used throughout the network, but will be morphed to cells with one functionality (such as fully connected, convolution, max pooling) later on after training. As a special case, one type of the cell disconnects the input from the output, which essentially changes the network structure. This is pretty much similar to the “stem cells” in humans.

In order to achieve that, there will be an encoding to the universal cells to give “preference” of the actual cells that they might morph to. Those encodings will be able to be mutated so that better networks can be identified. This is similar to the chromosome, encoding the entire functionality.

The above two advancements form the foundation of evolution, at which stage the AI is able to gradually escape human control.

What makes the AI really independent is the rediscovery of composition, only this time is by AI themselves. Once the AIs can freely compose/decompose, together with their learning capability, the AIs will scale exponentially and there is no way any human can catch up or control them.

That is the time that we will embrace a world dominated by AIs.

Will Humans be Taken Over by AI?

In short, yes. Humans are just one small stage in the long history of evolution. It has its time of prime, and has its time of decline[14].

[14] More discussion on the stages of evolution can be found in “Some scribble of things”.

The inefficiency of human beings

We have to realize the internal inefficiencies of humans, which is the main reason that humans will eventually be replaced by any being that is far more efficient, AI or not.

The fundamental inefficiency of human beings is the separation of learning and evolution. A human can learn a great deal of things in her lifetime. But after her death, nothing is left. Her offsprings do not possess any knowledge she has learned, and have to start from ground zero again. Evolution is always semi-random and selected by the environment, which is not controlled by human beings (currently). Granted, the human can leave traces of her life (such as books, videos) to the later generations. The humans can form societies, which are forms of compound life that span multiple generations. However, in my opinions, they are just fruitless efforts to be more efficient to compensate the fundamental limitation of the semi-random evolution[15]. On the other hand, AIs do not have this limitation, the boundary between learning and evolution is very fuzzy. The next generation AI can start from the learned AI of the current generation, which is much more efficient.

Humans are generalists. They can perform many tasks, but for each of the tasks, something (whether human built or not) can perform better. Humans are limited by its physical capability, the bone structure, body strength etc. The human's only advantage is her ability to create, to compose, and to predict. However, those capabilities will soon be learned by AI also. It is hard to believe that iron ion transporting across the brain cell membrane is a fast process. Once the same process can be reproduced in silicon, humans really possess no more advantage over AI.

When will the strong AI emerge?

As I have mentioned last year, intelligence is a composition of three factors: **perception**, **memory**, and **reasoning**. If we look back into evolutionary history, it takes billions of years to evolve perception, hundreds of millions of years to evolve memory, but it only takes hundreds of thousands of years to evolve reasoning. In this sense, reasoning is not difficult to evolve, given perception and memory, in this accelerated evolutionary process.

Over the past thousand years, the human's capabilities have not changed much, but knowledge and technology have advanced exponentially. This precisely implies that knowledge and technology development is not difficult, and are advancing at an exponential pace.

If we look back into the AI history, we have developed the ballpark of the perception; we are (less than) halfway in the process of developing memory, and it's just a matter of time that AI becomes capable of reasoning, no matter whether we develop it or not. Once they grasp reasoning, it is very easy for them to condense thousands of years of human knowledge and

[15] Nevertheless, this inefficiency also preserves the variation of the human race, which makes it less likely to be extinct in many scenarios.

technology in a short period of time, with much faster iteration. We cannot forget that AI can easily learn all the knowledge and technology humans have developed, which is much faster than exploring them from scratch themselves.

If everything is on the same trajectory, and follows the same pace as human's evolution, I'd say the creation of a strong AI is very soon (tens of years).

When will superintelligence emerge?

After the creation of human-level AI, when will AI be much smarter than human (superintelligence)? According to Tim Urban, the time would be really short, because AI can learn at an accelerated pace. It's just like a positive feedback loop, the smarter the AI is, the faster she learns, and the smarter she is the next moment.

I'm not that optimistic, however. Real life is not as simple as the game of go, whose rules can be concisely described and moves projected at a fast pace. Some learning cannot be done without really trying it out physically, which is limited by the physical resources at hand. Thus, simply by thinking and projecting cannot make a strong AI a superintelligence. In this sense, I don't believe that an AI working on perfecting handwriting is capable of annihilating the entire human race as a side product, only after hiding a month in the internet unnoticed, as Tim Urban has described.

I also don't believe that human-level AI can become superintelligence simply by learning. If so, humans (definitely human-level intelligence) are already superintelligence. The advantage of a human-level AI is that she may not die and lose everything she has learned. If she learns at the same pace as humans (she's no smarter, right?), it may take her centuries to see her advantage.

Thus, evolution is a necessary step for human-level AIs to be superintelligence. Granted, AIs may evolve at a much faster pace than humans, but this process is still much slower than learning. Also, at human-level, evolution is a number's game. One AI cannot become superintelligence alone.

The transition won't happen overnight, a few days, or even several months. However, I don't believe the timescale would be centuries either. If an AI does become a superintelligence, and is evil to humans, we sure will notice it and have time to react. It will be a long and painful battle and humans are almost certain to lose eventually, just like what the terminator movie projects.

What is the future of humans?

One may wonder whether humans and AI can co-exist in harmony. I'm not that optimistic.

Only one dominant species can exist. Humans are the current dominant species, and AI is the next. Some lower level animals, such as rabbits, rats, cockroaches, utilize the strategy of mass reproduction to ensure the survival of the race, which may well survive the change of the dominant species. The strategy of humans is to maintain dominance, which happens to be the same strategy of AI. Do you remember what humans do to the next-in-line dominant species? This is what AI will do to humans.

The dominance of AI is much riskier than any new dominant species evolved from humans, simply because AI is not even in the food chain. Life still recovers after each of the past five extinction events. One major reason is that no single higher level species can survive alone because every species is in the food chain. Not any more for AI. In the worst case, no carbon based life form will exist once AI dominates. This is an entirely uncharted territory.

Maybe the human's future really relies on the integration with AI, as Elon Musk projects. Only that would ensure the survival of a small percentage of the human population.

What Shall We Do Now?

Can we steer the AI to be benevolent? Can we implant something like "the three laws of robotics" to AI?

Well, it is difficult, but not impossible. The speed of AI training itself is so fast that it is difficult to steer its direction. In addition, a key step to superintelligence is evolution, which undermines the effectiveness of any steering or implantation. The three laws of robotics are fundamentally against the first principle and will not have any long lasting effect. However, it doesn't mean we don't need to do anything now. We still need to try steering or implanting something smarter and pray for a benevolent superintelligence on our side.

For me, I'm determined to plant a bug in the AI, or die trying. I don't trust anyone else on this task.

Buckle up. We are in the dawn of great change. The world will never be the same.

Fei Sun@ San Jose

v0.1.1 - Mar. 2021: minor grammar edits and typo fixes
V 0.1 - Dec. 2017: initial draft