Artificial Neural Networks, Genetic Algorithms and Quantum Computing

An Exploration into Cutting Edge Computer Science

Nathan Hillyer
Department of Computer Science
Lewis University
Romeoville, IL
nathan@hillyer.me

***Abstract*—This white paper addresses the state of neural networks, genetic algorithms, and quantum computing.**

Keywords—computer science; neural networks; genetic algorithms; quantum computing; artificial intelligence

I. Introduction

Artificial neural networks, genetic algorithms and quantum computing represent three promising areas of computer science research.

II. Artificial Neural Networks

An artificial neural network, in Computer Science, is a model for developing computing systems or algorithms that attempt to replicate the human brain's architecture. Computers have yet to reach the level of a human brain's general intelligence and ability to solve novel problems. An artificial neural network tries to bridge this gap by emulating the human brain's method of solving problems. The complexity of many human brain feats are often taken for granted. Our brains are capable of recognizing faces, ascribing entire mood changes based on subtle changes within those faces, and telling the difference between a cat and a dog. These are computationally demanding feats that have proven difficult for computers to replicate. Another advantage of the human brain is that, if you take into account all approximately 10 billion neurons, the brain is a highly parallel system. Computers have difficulty with highly parallel processing—so much that an entire field of study (distributed algorithms) is dedicated to harnessing the power of parallel computation. The human brain is able to infer conclusions based on imprecise information, but computers have primarily excelled at solving problems with highly specific information. The strengths of the human brain over traditional models of computation gave rise to artificial neural networks.

A distinguishing feature of artificial neural networks is their ability to learn over time. This could be juxtaposed to the problem solving approach of an expert system where many conditions are preprogrammed into the system ahead of time in an attempt to mimic intelligence. For example, a chatbot could be programmed conditionally to respond to the prompt "How is the weather today?" with many responses considered to be reasonable answers by the programmer. An approach using artificial neural networks, on the other hand, could take many potential responses and couple them with a data source, such as how human-like the person on the other end considered the response, and try to predict how well a newly generated response may perform in a Turing Test based on the similarity of its sentence structure to high performing responses.

A. Example Uses of Artificial Neural Networks

Artificial neural networks have had a number of successful practical applications. We will cover a couple broad use cases coupled with a specific example and leave it an exercise to the reader to envision the myriad of additional possible applications.

They have been used with success with classification, a form of pattern recognition where a set of observations are used to tune an algorithm that best fits those observations. An example would be an email spam filter that is trained by actual data of people indicating they think an email is spam. A major advantage of using an artificial neural network to classify data is it's ability to become smarter as more data is made available.

Neural networks have also been used with success in making predications based on observations. One example would be their use within the financial industry as tools for stock market prediction. Neural networks have been claimed to have as much as a 199% percent return in a 2-year period with the usage of artificial neural networks.

Artificial neural networks have also seen limited success when applied to the field of robotics. Prediction and classification are two major hurdles when attempting to create an intelligent robot. For example, a robot needs to be able to make reasonable predictions when mobility comes into play to be able to successfully and intuitively navigate its environment. Classification is also important, as successfully navigating an environment involves correctly classifying what one is perceiving in order to formulate parameters of possible action. One reason the success in robotics has been fairly limited tis that the setup and training of a neural network takes a lot of time and thought, consequently it isn't easy to have it apply to a dynamic rapidly changing data source such as incoming perception.

B. Limitations of Artificial Neural Networks

A large limitation of artificial neural networks is how much time and effort it takes to create and maintain an artificial neural network. Even a fairly simple neural network, such as one that tries to predict a persons test score based on how long they've slept and studies, takes a number of complex mathematical computations before one can arrive at a suitable algorithm. The algorithmic performance of artificial neural networks is another area that can be problematic. One mathematician has said, "Although neural nets do solve a few toy problems, their powers of computation are so limited that I am surprised anyone takes them seriously as a general problem-solving tool".[1] Given that this quote was given in 1997, it is perhaps reasonable to assume this performance limitation has been assuaged over time as computing power continues to nearly double every couple years. However, this doesn't change the fact that neural networks usually require a large amount of computations, especially for large data sets, before arriving at solution. This brings us to our next point, in that artificial neural networks require a large training data set in order to have a reasonable degree of accuracy. This makes their application limited in more novel areas where data isn't yet readily available—for example, if one was trying to create the first email spam filter, without any prior data on what constitutes spam, an artificial neural network wouldn't be a great approach.

III. Genetic Algorithms

Genetic algorithms attempt to mimic the process of natural selection to find an optimized solution to a problem. A genetic algorithm takes a set of possible solutions as its input and these solutions have the ability to be changed and combined together. These solutions are combined with a function that determines their fitness—how well they solve the problem—to mimic natural selection. The first generation of solutions attempts to combine the best solutions to create another generation, and so on, until an optimal solution is found.

Genetic algorithm can be understood by evaluating the phases they go through. The first phase is initialization, where the creator of the algorithm creates many possible solutions to the problem. The second phase is the evaluation of these solutions based on our fitness function. The third phase is the selection of candidates for the next generation of solutions—basically we discard the solutions that perform poorly. The fourth phase is combining these solutions together to create new solutions—our next generation. The last phase is mutation, where we introduce changes at random to our solutions in the hope of creating a high performing novel combination. These five phases are then repeated as long as necessary based on time constraints and performance parameters and then a solution is picked.

A. Example Uses of Genetic Algorithms

Genetic algorithms have been used in robotics to teach robots how to learn various activities, such as how to walk.

Genetic algorithms are often used for scheduling problems. They can take a set of constraints, such as time and resources and come up with an optimal schedule for something like a factory floor.

Genetic algorithms are used for complex search problems, for example they could be used to find the best combination of materials to create a compromise between strength, weight, and cost for a drone manufacture.

B. Limitations of Genetic Algorithms

The requirement of a fitness function often limits the potential use of genetic algorithms. Not every problem domain has a neat way of determining if a solution is optimal. Another problem with genetic algorithms are their propensity to scale poorly. Because each generation results in another iteration, it takes a lot of processing to get through large data sets that end up requiring many generations before finding an optimal solution. Another problem is that genetic algorithms can come to a local optimum solution where it appears that the solution has settled at the best possible, but, in reality, if n more generations were ran there was a much better solution.

IV. Quantum Computing

Quantum computing is an attempt to create applications of discoveries within quantum physics to the field of Computer Science. The main difference between a quantum computer and a regular one is that a normal computer has memory consisting of bits—an on or off state, 0 or 1. A quantum computer, on the other hand, stores qubits, which can be 0, 1, or a superposition between 0 and 1. The end result of this uncertainty is that a quantum computer can be massively parallelized, because of the number of possible states multiple qubits could represent. Quantum computing is still a mostly theoretical concept, as there haven't been any large scale quantum computers fulfilling the promises put our by quantum theorists.

A. Example Uses of Quantum Computing

As quantum computers are only available in a very early form, there are few practical and proven applications of quantum computing. However, the theory has many possible uses should the full potential of quantum computing be harnessed. The most famous potential use case would be decrypting public key cryptography systems. There has been an algorithm (Shor's algorithm) written in 1994 that solves integer factorization in polynomial time. Integer factorization is a key component of public key cryptography and should this algorithm ever be successfully be implemented in a full scale system, the effect would be drastic as it would nullify the Internet's most common solution for keeping information private and secure.

Another algorithm, Grover's algorithm, has been written for quantum computers. This algorithm could be implemented as a more efficient way of searching a database when compared to classical algorithms.

B. Limitations of Quantum Computing

The primary limitation is that there has yet to be an quantum computer implementation that demonstrates the speedup demonstrated by theorists. There have been quantum computers made, but because they haven't yet fulfilled the promise, there are still many open questions as to whether or not the theoretical underpinnings of the field are sound.

References

1. Dewdney, A.K. (1997). Yes, We Have No Neutrons: An Eye-Opening Tour through the Twists and Turns of Bad Science. New York: Wiley. p. 82.