Complexity Complexity: Intuition and Abstraction to Computation, Probability and Black Boxes A Review

Tyson A. Neuroth
UC Davis, Computer Science Departmnent
taneuroth@ucdavis.edu
Physics 256 Final Report, Professor James P. Crutchfield

2018

Abstract

In this paper, measures of complexity from algorithmic information theory are reviewed and discussed in relation to the physical world, and the data generated by it. Recent attempts to further our theoretical understanding of how neural networks successfully process complex information have inspired a closer look at the potential ways in which complexity informs the optimal topology of neural networks, in particular their depths. One of these ideas relates measures from algorithmic information theory, in particular logical depth and Kolmogorov complexity to the depth in a neural network. Since these algorthmic measures are not feasibly computed, I propose to study the respectively similar measures, statistical complexity (C_μ) and entropy rate (h_μ), which can be easily computed. Throughout the review, a major theme that is explore is the idea that people manage complexity in much the same way that a neural network is able to, through generalization and abstraction.

1 Introduction

Complexity science is perhaps one of the most fundamentally human endeavors, applicable to nearly all intellectual pursuits. This is a natural consequence of our existence, as thinking beings living in a mysterious and complicated world. While we are arguably optimized to live and function in an environment, we also collectively set out a diverse distribution of quests to demystify existence, and build on a deeper knowledge from what we can observe and deduce. Ideally, we can one day be (Keano Reeves)-like. But this situation invokes an uncomfortable question. Are we even capable to comprehend the knowledge that we seek? We can be sure that all of the details are even beyond the storage capacity of the human brain, and all of the logical branches likely beyond the number of neural pathways. Yet how much of it is redundant and how much of it is random? How much can be simplified, abstracted, and generalized? The answers to these questions not only determine the feasibility of our ultimate intellectual quests, but also the paths that we can take to fulfill them. Aptly the word complexity has become a grand keyword, the quintessential semantical super term, appealing to our intuition and intellectual identity, and entrusted to inform us, vaguely, to that which we are getting ourselves into. In turn, the idea of complexity itself can be considered highly complex. That is, the super term is non-trivially super.

Alongside our quests for understanding, other more immediately practical endeavors are abound. In particular, we are concerned with how to make optimal decisions under a large of array of contexts. In the current information age, data has naturally become a prime commodity which gives one the ingredients to predict future events. Yet the amount of data, its diversity, and the daunting levels of complexity in the processes that produce it lead to a major dilemma. Our brains are not equipped to process it all in its basic form.

In the earlier days, artificial intelligence was based more in logic. People imagined to design machines that would simply deduce in a manner that people do. The problem turned out to be that to design such a machine, we sort of need to know how to the the deductions should be made. Thus our own logical and mathematical capabilities would limit those of the machines we create. Later, many breakthroughs were made for practical artificial intelligence with a more statistical basis. By learning relationships more implicitly, we were still able to gain actionable insight into how to direct our future. By now, many of the most successful methods are mysterious black box tools, such as deep neural networks. While these tools satisfy many impulsive needs, they ultimately leave us in an unsatisfying intellectual situation. We don't really know what is going on. Even worse, the realization of a working solution demotivates an effort to seek a more explicit solution that would advance our understanding and better inform the design of future solutions. The risk, is that human intellect eventually becomes largely obsolete, and we go into the future simply as passengers with little intuition about why or where we are going.

Perhaps a ray of sunlight is that it would seam that these successful black box data processors might be relying largely on the same tools that human beings use to make sense of their environments. In particular, it would seam that abstraction, and generalization are a key component to learning about many of the large and deeply complex objects that exist in our universe. And most interestingly for us, there may be simple fundamental laws that govern the ways that information can be simplified, structured, abstracted, and learned from, as well as how it flows through time and propagates.

In this text I briefly complexity with a focus on algorithmic information theory and the possibility that insights gained from theoretical computer science can generalize to the physical realm and the data produced by it. Next, I discuss some recent work relating algorithmic information theory to deep learning, as well as a few other papers about the inner workings of deep neural networks. Finally, I propose an experiment aimed to find empirical evidence to relationships between the statistical complexity and entropy rate of an information source, and the depth and structure of a well performing neural network for predicting it.

2 Background

Over the years, quite a lot of complexity measures have been introduced. Llyd provided a non-comprehensive list of some of the more general measures, which he organized into 3 roughly appropriate categories: difficulty of description, difficulty of creation , and degree of organization. Another common distinction is simply that between order and chaos, or structure and randomness. Yet, it has become clear that the two are not independent and there is more going on in-between. Some extraordinary things in particular happen right at the onset of chaos [7]. This type of phenomena, where information or information generating mechanisms can be considered to over-go phase transitions, should be strong motivation for further research for more examples, and generalizations. Besides research in dynamical systems, some interesting results have been found that suggest a possible universality of phase transitions in high-dimensional geometry and data modelling [12]. Another possibly related are of mathematical research is based on Ramsey theory, which is concerned with for example, how properties emerge as the size of a set grows. One example is the Pythagorean triple problem [10].

Alongside mathematical results, Crutchfield has written many papers which include a philosophical perspective about notions and implications of complexity, chaos, and information which are relevant to this paper

[9, 6].

A further distinction between types of complexity measures is those with a basis in first order logic or computation, which tend to assume a more deterministic setting but are usually not feasible to compute or accurately estimate, and those based in statistics and probability. Some discussion on this distinction and in particular about the nature and usefulness of statistical complexity measures can be found in [13].

This paper focuses on algorithmic information theory, which is based in the theoretical formalization of computing pioneered by Alonzo Church and Alan Turing, with the specific model of computation assumed to be a Turing machine.

2.1 Turing Machine

Alan Turing proposed the Turing machine as a model that closely resembles the way a human would perform a calculation. Assuming that one had and unbounded supply of time, paper, pencils and erasers, a being could theoretically calculate anything which can be calculated. Formally, the notion is that the Turing machine can compute any computable function of the natural numbers, and this is the assumption held by the Church-Turing thesis, which is the foundation to nearly all of theoretical computer science. To avoid overwhelming with detail, a formal definition is not given here, but can be easily found on Wikipedia, for example. The important characteristics are that the Turing machine has a finite set of states, and an unbounded amount of memory. A function not computable by a Turing machine may be such that one simply cannot determine if a solution exists without searching over an infinite space. That is, Turing machines are allowed an unbounded amount of memory and time, but not an infinite amount. The input of a Turing machine is the initial sequence of discrete symbols on its tape (analogous to its memory). Since any finite discrete object can be encoded as a string of symbols (even binary), it is often assumed for simplicity that the alphabet is binary. Throughout this paper, the term program will be used, and it is assumed that the program is in the form of a Turing machine.

2.2 Kolmogorov Complexity

The Kolmogorov complexity of string, or discrete object, is the size of the smallest possible Turing machine that generates it. The Turing machine can be considered an idealized compression algorithm that need not apply to other objects. In the worst case, the Turing machine has the object itself embedded in it, along with a print program. Ming et al. give a larger overview of Kolmogorov complexity and its applications [17].

2.3 Logical Depth

Charles Bennett introduced logical depth, as a measure to capture the intuitive notions of organized complexity[3]. In particular, Bennett suggest that some objects show "evidence of a non-trivial causal history". The idea is that each object, in this case encoded as a string, requires some minimal number of steps to create by a plausible process. Since the print program itself could be the process, which has a run-time always near linear with respect to the size of the object, some restrictions are needed to give logical depth useful meaning. Bennett refers to results and assumptions in scientific reasoning and inductive inference [25] to support his assumption that the simplest process is the most plausible since it is the most economical. Thus he invokes Kolmogorov complexity and formalizes logical depth as the number of steps taken by the smallest program to computes the object. Just as Kolmogorov can be related to compressible of an object, Logical depth can be related to the cost to perform its decompression. Of particular concern with logical depth, is the dependence on the program being the absolute smallest program since it can be the case that even a slightly larger allowance can result in a wildly different measurement of depth. This issue is discussed as an issue of stability with respect to logical depth being consistent with the intuitive notions that it aims to capture. In other words, if randomly adding a single character to a sting drastically changes its measurement of depth, then it is hard to argue that this measurement is capturing the intuitive notion of depth. Another issue of concern, is the dependence on the computational model. In addition to defining Logical depth for deterministic Turing machines, he also gave another definition for probabilistically deep strings based on a probabilistic turing machine.

2.4 Effective Complexity

Effective complexity was motivated by doubts as to whether Kolmogorov complexity is a good measure for capturing intuitive notions of complexity [14]. One may assume that randomly generated strings may have a large Kolmogorov complexity, yet a completely random object is not generally considered to be the most complex. Effective complexity therefore aims to measure the complexity of the non-random parts of the string, which is found to be essentially the same as the logical depth ignoring the random parts. Intuitively this is a good approach, but since Kolmogorov complexity and logical depth are primarily considered in deterministic settings, where the object itself is described explicitly, and not as simply a realization of a probabilistic generator, it is not clear how one can assume that any part of the object is random. Additionally, it is hard to argue that

a random looking string necessarily has a large Kolmogorov complexity. As an example, I wonder if a such a string might be the result of an easy to describe yet computationally difficult problem. As an example, a very large random looking string might be an encoding of the smallest solution to a simple diophantine equation such as $w^5 = x^5 + y^5 + z^5$; the program to find it can simply be a brute force search algorithm, which should be small, yet the answer could be very large. In fact, it is not yet known if $w^5 = x^5 + y^5 + z^5$ has a solution at all.

Nevertheless, the issue of distinguishing randomness from order seams to be an important one to resolve since it is presumed that random processes are fundamentally involved in the evolution of the physical universe. One can hope that we will eventually be able to better unify what we have learned from algorithmic information theory and computation, with what we know about real world stochastic processes. But it is not certain that it would work out smoothly.

2.5 Can Algorithmic Complexity Help Describe Physical Reality?

Besides the issue that complexity measures such as Kolmogorov complexity and logical depth are not computable, there are doubts among some that these measures can truely describe our reality. Mueller suggests that this issue is part of the larger question which asks whether the physical world is fundamental or emergent [18]. It is shown that computational mechanisms can evolve and emerge in physical systems [8]. The question is whether ultimately there are simple fundamental laws the govern computation in the physical world at the lowest levels. Boyde and Crutchfield have published on fundamental thermodynamic costs for information storage [4]. At even lower levels, quantum physics is involved in information dynamics and natural computation. Besides the head scratch inducing phenomena of quantum strangeness, the quantum real prompts us to question reality further, including instinsicality of classical laws in computational complexity theory, as well as other complexity measures. As an example, Aghamohammadi et al. have recently demonstrated that systems may appear more or less simple relative to others depending on whether they are represented through classical physical descriptions or quantum descriptions [1].

3 Abstraction and Complexity

A idea that this text is working towards arguing, is that, in essence, abstraction and generalization are core to the inner workings of deep black box learning models, in addition to human learning and reasoning. Saitta et al. explored these concepts for general artificial intelligence [20]. Assuming a particular model of abstraction, based on their $\mathcal{KRA}$ framework, they found evidence that abstraction is related to complexity measures, but not monotonically correlated. A book authored by Saitta and Zucker covers a range of related topics, from language and mathematics to complexity measures, abstraction and artificial intelligence [21].

4 Neural Networks and Deep Learning

Neural networks became popular with the increase in computational power and access to large amounts of data. Now it has become a primary tool for modern technologies, such as computer vision, and speech recognition. While successful, the lack of theory explaining how they work is a continuous source of frustration. In particular, it is not robustly understood what is the role of depth (multiple layers) in determining their effectiveness for different prediction problems. It is generally understood that the properties of the data are an important factor. Thus, it is natural that measures of complexity become a major part of our attempts to reason about how the neural network is leveraging depth to effectively make predictions from the data.

4.1 Network Depth, Optimality and Training Dynamics

Modern definitions for depth in neural networks are based on concept of credit assignment, a problem arising in reinforcement learning [24, 15]. The credit assignment problem asks how credit should be assigned/divided between the different choices that are made for a successful realization of a particular unfolding of events with a final conclusion. For example, one may ask how important a particular chess move was to the ultimate victory over an opponent. In the context of neural networks, Schmidhuber popularized the term credit assignment path (CAP) as a measure of the depth of a neural network. The CAP is formalized through the description of hidden layers in the network as weighted predicate functions. The rough idea is that each layer makes a sort of weighted combination of decision, with implications propagating forward to subsequent layers. Problem depth is then defined as the minimal CAP to solve a particular problem [22]. The depth necessary to optimally predict should then be based on the problem depth. Unfortunately, like logical depth, this is not a computable measure.

In fact, it is not precisely clear why depth (more than 1 hidden layer) is needed at all, since by the universal approximation theorem[11], a single hidden layer should be enough to successfully approximate almost any continuous function from the input layer to the output layer. The important catch is that the network also

needs to learn the weights that form the function. The algorithm that is used, is called back propagation, which iteratively attributes portions of the error to nodes in previous layers, to adjust their weights while searching for a good local optimum in the solution space.

Besides the structure of the network, learnability is a highly interesting topic of study. It has even been shown that the training process can be analyzed as a dynamical system that can enter chaotic regimes[19].

4.2 Algorithmic Information Theoretic Views

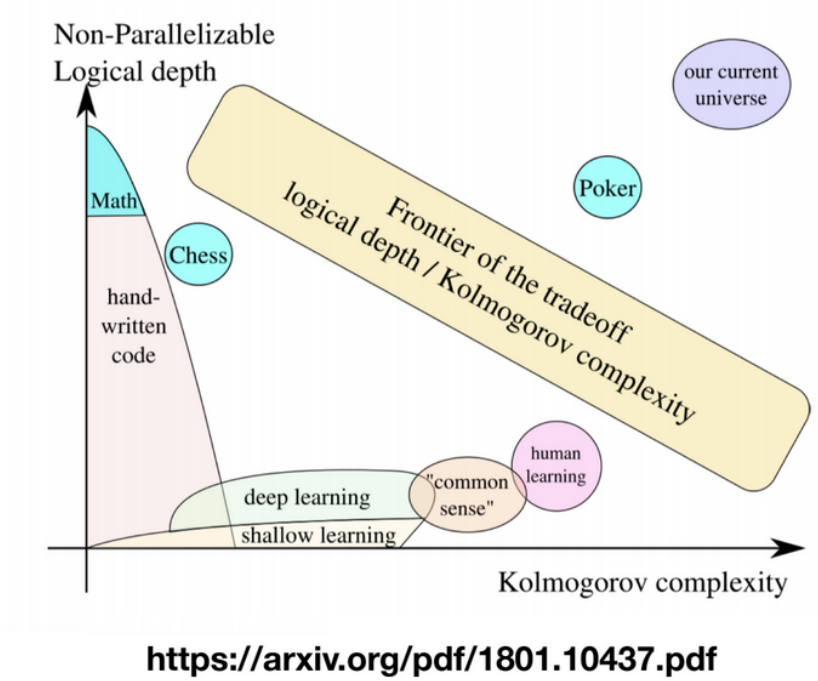

Figure 1: Different Objects placed in the space of Kolmogrov complexity and Non-Parallelizable Logical depth according to Lê Nguyên Hoang, Rachid Guerraoui. The image sourced from their paper, "Deep Learning Works in Practice. But Does it Work in Theory?", https://arxiv.org/abs/1801.10437

Recently, another conjecture is made that logical depth underlies the necessity of depth in a network [16]. They reviewed some interesting ideas and insights that trace back to the early days of theory of computation from Alan Turing and others. In particular they consider the complexities of the human brain itself, according to the number of neurons and their connections, and contrast this with they capabilities and limitations for humans to write codes or construct algorithms capable of matching such complexity. They suggest that the depth of a neural network is necessitate by the logical depth of the objects it needs to successfully reason about. In particular, they argue that real world objects typically have large logical depth. Figure 1 shows their diagram where different objects are roughly placed in K-complexity-Logical Depth space.

4.3 Neural Networks and Abstraction

Researchers have been trying to theorize about the inner workings of neural networks using information theory [5]. In particular, the question is how information flows through a neural network. The nodes and layers are considered random variables, and the flow of information is conceptualized through concepts such as conditional entropy and mutual information.

Evidence has been building that a major key to the success of a neural network, is its ability to use some sort of abstraction, or generalization, for simplifying a highly complex problem. This is in fact the same way that people seams to be able to make sense of their environments and manage complexity. Researchers have published results as evidence of this phenomena in action within neural networks [23], with an information theoretic basis. The idea is that there is an information bottleneck, which limits how much information can "feasibly" pass through the layers between the input and the output. In order to squeeze more information through, the first hidden layers supposedly do a sort of generalization, by abstracting related parts of the data into features, and leaving the lesser important details behind. In essences this is a process by which the information undergoes

a type of lossy compression to pass through the bottleneck, and afterward it is decompressed and decoded, so to speak, before reaching the output layer. Under the right type of conditions, topology of the network and activation functions, along with a good training algorithm, hopefully this mechanism would be achieved. Like people, it seams that our neural network training algorithms cannot sometimes not effectively learn from raw data when it has a certain level of overwhelming complexity. An interesting question would be, are there rules we can find that tell us about the thresholds and limitations on information complexities that necessitate certain levels of "lossy" abstraction.

Ideas stemming from the concepts of logical depth and Kolmogorov complexity may actually be useful for reasoning about these issues.

4.4 Entropy Rate and Statistical Complexity

While the idea that logical depth and Kolmogorov complexity could be useful heuristics for optimal network topology construction, the fact that they are uncomputable makes this a limited and difficult path for further study. However, since statistical measures of complexity are practical in application, it may be possible to find new relationships between them and the depth and width of an effective neural network. By starting with an ϵ-machine, one can know a priori many complexity measures of the process that it models. In particular, it can be argued that the statistical complexity, C_μ of a process is analogous to the logical depth of an object. After all, Bennett's original claim was that depth should be a rough measure of the non-trivialness of the objects causal history. This is essentially what statistical complexity measurses, since it is the entropy of the asymptotic distribution over its causal states. Another measure, entropy rate (h_μ), is very closely related to Kolmogorov complexity. Thus, these two measures seam to be good choices for future investigation in their relationships to neural network topologies.

4.4.1 Feed Forward Network for Predicting a Binary Time Series

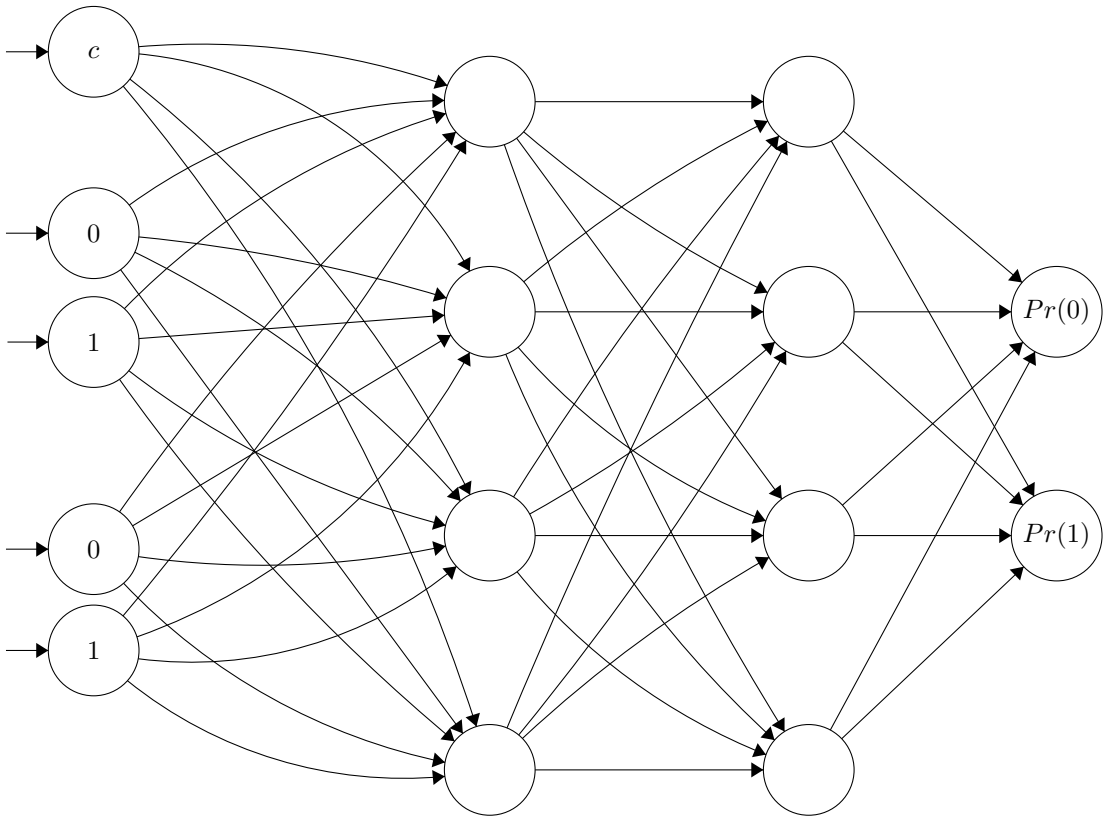

Figure 2: A depiction of a simple feed forward neural network for predicting a next symbol from a finite past of a binary time series. The inputs use one-hot encoding.

While the relationships between C_μ, h_μ and neural network depth are testable in theory. Designing a controlled study is difficult, since many compounding factors can affect the results. To simplify the study I propose that a simple feed forward neural network be used as depicted in Figure 2. The input layer uses one-hot

encoding to take in discrete binary inputs. That is, there are two inputs for each symbol, and if the symbol is a 1, then one of the inputs is hot (1), else the other is hot. The network takes in some fixed amount of the past and predicts the next symbol. It is trained over a large realization chain, that is generated by the ϵ-machine.

However, even using this simple model, there are many important considerations. The objective is to study the relationships with depth (number of hidden layers), yet each hidden layer can also have different widths, and this aspect is important as well for optimizing the neural network structure. In addition, different processes could have different Markov orders and require different amounts of the past to synchronize and optimally predict the future. Furthermore, it is not clear how including more than the necessary amount of the past as input would affect the training, and how the internal structure depends on these factors as well. Since we wish to see how well the structure is optimized, we should also normalize the performance according to what is theoretically possible, which is easily determined from the ϵ-machine based on the excess entropy.

A basic approach would be to test sets of ϵ-machines which have very similar properties, up to C_μ and h_μ, which would be varied. Then one could devise a systematic approach to sample ϵ-machines in this complexity space with these constraints. Then a fixed amount of the past (input layer size) can be used to predict. Simple rules that govern the freedom to varying the widths of internal layers, in addition to varying the depth need to be carefully formed. A naive approach to start, is to keep the width fixed. Next, one could use approximation methods for optimizing the structure to best predict the data. For example, simulated annealing (a randomized algorithm for approximating an optimal solution) has been studied for network optimization [26, 2].

I have developed some code for this study, but it remains incomplete. I hope to find time to follow through, and will share the results when they become available.

5 Complexity Complexity

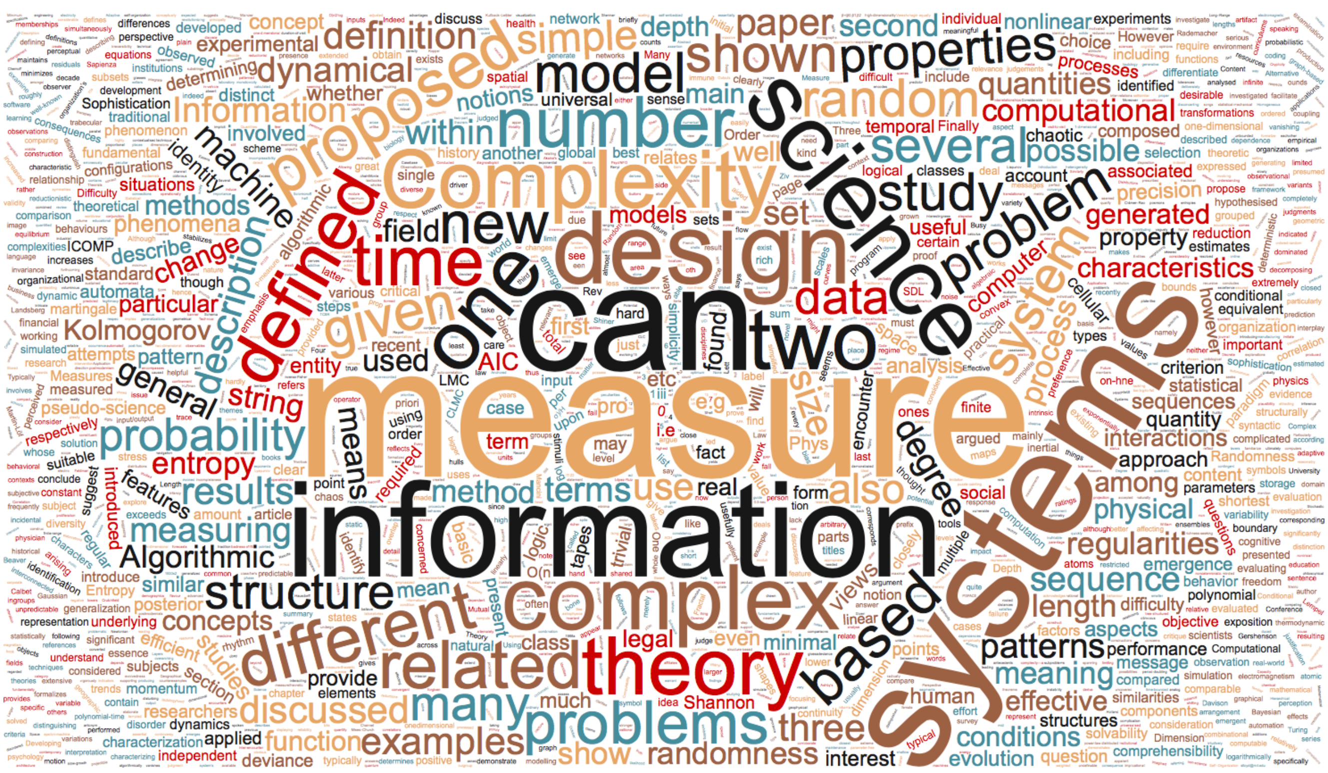

Figure 3: Caption

A major theme that I indented to explore, is that complexity in and of itself (as a concept) is not something simple. One of my thoughts on the subject is as follows. To manage real world complexity, we try to generalize and abstract as much as necessary, to reason about things which would be far beyond our ability if viewed in raw form. To do this appropriately, we need to be able to make reasonable assumptions. At the same time, as we ignore details, and begin to think more about different things through generalized abstractions, the number of assumptions, restrictions and special cases that we need to consider increases; presumably at some point to a level where the application of the generalization becomes either too meaningless, or too complex itself

to manage. While a lot has already been done, and it's not an easy topic, the most interesting direction for future research in my opinion, is to learn more about these types of trade offs. And in general, I would like to know more about the ways that information can be structured and random, and how these properties are fundamentally connected, and limited.

Figure 3 shows a word cloud that I generated (using the tool at www.wordclouds.com) based on the text in a large number of research paper abstracts about complexity in general, and in specific domain applications. This illustrates the large number of notions that we aim to capture with complexity measures. Without further comment, I leave the image up to interpretation.

References

[1] Cina Aghamohammadi, John R Mahoney, and James P Crutchfield. The ambiguity of simplicity in quantum and classical simulation. *Physics letters A*, 381(14):1223–1227, 2017.

[2] Vina Ayumi, LM Rasdi Rere, Mohamad Ivan Fanany, and Aniati Murni Arymurthy. Optimization of convolutional neural network using microcanonical annealing algorithm. In *Advanced Computer Science and Information Systems (ICACSIS), 2016 International Conference on*, pages 506–511. IEEE, 2016.

[3] Charles H Bennett. Logical depth and physical complexity. *The Universal Turing Machine A Half-Century Survey*, pages 207–235, 1995.

[4] Alexander Blades Boyd. *Thermodynamics of Correlations and Structure in Information Engines*. PhD thesis, University of California Davis, 2018.

[5] Ahmad Chaddad, Behnaz Naisiri, Marco Pedersoli, Eric Granger, Christian Desrosiers, and Matthew Toews. Modeling information flow through deep neural networks. *arXiv preprint arXiv:1712.00003*, 2017.

[6] James P Crutchfield. Knowledge and meaning: Chaos and complexity. In *Modeling complex phenomena*, pages 66–101. Springer, 1992.

[7] James P Crutchfield. What lies between order and chaos? In *Art and complexity*, pages 31–45. Elsevier, 2003.

[8] James P Crutchfield and Melanie Mitchell. The evolution of emergent computation. *Proceedings of the National Academy of Sciences*, 92(23):10742–10746, 1995.

[9] JP Crutchfield. Chaos and complexity. *Handbook of Metaphysics and Ontology*, 1990.

[10] Luís Cruz-Filipe and Peter Schneider-Kamp. Formally proving the boolean pythagorean triples conjecture. In *LPAR-21: 21st International Conference on Logic for Programming, Artificial Intelligence and Reasoning*, volume 46, pages 509–522, 2017.

[11] Balázs Csanád Csáji. Approximation with artificial neural networks. *Faculty of Sciences, Etvs Lornd University, Hungary*, 24:48, 2001.

[12] David Donoho and Jared Tanner. Observed universality of phase transitions in high-dimensional geometry, with implications for modern data analysis and signal processing. *Philosophical Transactions of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, 367(1906):4273–4293, 2009.

[13] David P Feldman and James P Crutchfield. Measures of statistical complexity: Why? *Physics Letters A*, 238(4-5):244–252, 1998.

[14] Murray Gell-Mann and Seth Lloyd. Information measures, effective complexity, and total information. *Complexity*, 2(1):44–52, 1996.

[15] John J Grefenstette. Credit assignment in rule discovery systems based on genetic algorithms. *Machine Learning*, 3(2-3):225–245, 1988.

[16] Lê Nguyên Hoang and Rachid Guerraoui. Deep learning works in practice. but does it work in theory? *arXiv preprint arXiv:1801.10437*, 2018.

[17] Li Ming and Paul MB Vitányi. Kolmogorov complexity and its applications. In *Algorithms and Complexity*, pages 187–254. Elsevier, 1990.

[18] Markus P Mueller. Could the physical world be emergent instead of fundamental, and why should we ask?(short version). *arXiv preprint arXiv:1712.01816*, 2017.

[19] Razvan Pascanu, Tomas Mikolov, and Yoshua Bengio. On the difficulty of training recurrent neural networks. In *International Conference on Machine Learning*, pages 1310–1318, 2013.

[20] Lorenza Saitta and Jean-Daniel Zucker. Abstraction and complexity measures. In *International Symposium on Abstraction, Reformulation, and Approximation*, pages 375–390. Springer, 2007.

[21] Lorenza Saitta and Jean-Daniel Zucker. *Abstraction in artificial intelligence and complex systems*, volume 456. Springer, 2013.

[22] Jürgen Schmidhuber. Deep learning in neural networks: An overview. *Neural networks*, 61:85–117, 2015.

[23] Ravid Shwartz-Ziv and Naftali Tishby. Opening the black box of deep neural networks via information. *arXiv preprint arXiv:1703.00810*, 2017.

[24] Derek Sleeman, Pat Langley, and Tom M Mitchell. Learning from solution paths: An approach to the credit assignment problem. *AI Magazine*, 3(2):48, 1982.

[25] Ray J Solomonoff. A formal theory of inductive inference. part i. *Information and control*, 7(1):1–22, 1964.

[26] A Yamazaki, MCP De Souto, and TB Ludermir. Optimization of neural network weights and architectures for odor recognition using simulated annealing. In *Neural Networks, 2002. IJCNN'02. Proceedings of the 2002 International Joint Conference on*, volume 1, pages 547–552. IEEE, 2002.