A Neural Network Search Engine

(Technical Paper)

Deep Learning Considered Harmful?

(STS Paper)

A Thesis Prospectus Submitted to:

The Faculty of the School of Engineering and Applied Science
University of Virginia • Charlottesville, Virginia

In Partial Fulfillment of the Requirements of the Degree
Bachelor of Science, School of Engineering

Michael A. Klaczynski
Fall 2019

On my honor as a student at the University of Virginia, I have neither given nor received aid on this assignment as defined by the Honor Guidelines for Thesis-Related Assignments.

Signature____ Date __________

Michael A. Klaczynski

Approved ____ Date __________

Hongning Wang, Department of Computer Science

Approved ____ Date __________

S. Travis Elliott, Department of Engineering and Society

Introduction

Over the course of the past decade, deep learning has proven to be an incredibly useful tool for a variety of tasks, including image classification, speech recognition, and language modeling. Because of this usefulness, much time, money and effort goes into generating new deep learning-powered algorithms. In the technical portion of the thesis, we propose a method of reusing old algorithms, such that the cost of creating new algorithms through deep learning can be lessened. In the STS research paper, we will question the dominance that deep learning has taken over various fields of AI, and whether or not it will hurt those fields in the long run.

Technical Problem

In this research we intend to build a "deep learning search engine" for use by machine learning algorithms. Using unsupervised learning methods to find common deep learning structures, it may be possible to create an index mapping small structures to larger ones. Networks recognizing smaller structures taking shape could then potentially save time in gradient descent by attempting to apply and fit common solutions.

Rational

An inspiration for this idea is analogical reasoning in humans. An analogy is an "inference that if two or more things agree with one another in some respects they will probably agree in others" (Woolf 1971). Given knowledge of one system, a human can make guesses about an unrelated yet structurally similar system (Kedar-Cabelli 1988). Using analogical reasoning it is often possible to find a solution to a problem much more quickly, as it allows humans to skip a lot of trial and error learning.

Deep learning algorithms don't do this. While a data scientist might be able to guess from the data what shape of network would be most useful, the internal weights of the network need to be learned from scratch every time. Wouldn't it be nice if we could guess at these weights given previous neural networks? Consider the case of convolutional neural networks. A convolutional layer can be seen as a dense layer where the vast majority of weights are set to zero, and the ones that aren't follow a certain repeating pattern. Knowing that this structure is useful, we can often replace an entire dense layer's worth of learning with a much smaller set of matrices. Perhaps there are many structures like this hidden within trained algorithms, waiting to be exploited.

Beyond simple machine learning, this research has implications for other fields of AI. Having a library full of common functional components could help solve the "cold start" problem for some general intelligence systems, allowing them to quickly build their own low-level sensory functions without relying on separate perception layers.

Methodology

The research can be divided into four subsections-- Data Collection, Pattern Mining, Indexing, and Search.

Data Collection.

Before we can do anything useful, we need a data set representative of a large variety of deep neural networks. There are, fortunately, multiple libraries of pre-trained networks. Modelzoo.co, for example, has a set of over one thousand pre-trained deep learning models for a variety of deep learning frameworks such as TensorFlow, PyTorch and Keras. These may all have to be converted to a standard representation, such as ONNX (Open Neural Network Exchange) before use.

A potential problem with this method of data collection is that libraries of pre-trained networks may not be truly representative of algorithms in the wild. They are often intended as starting points on which to build a more complex system. By only considering generic, reusable algorithms we may be missing out on structures found in more specialized networks. Nonetheless, as a proof of concept, this will probably be adequate.

Pattern Mining.

There are many approaches to pattern mining, but here we will focus on frequent subtree mining. This approach doesn't find looping patterns, but runs far faster than more general graph mining algorithms (Chi 2005). Given that neural networks generally flow in one direction, the loops should not be too much of a concern. There are a multitude of subtree mining algorithms, and selecting the most useful one for our purposes will be part of the research.

By running our pattern mining algorithm on our collection of neural networks, we will create a database of common neural network patterns.

Indexing.

This is where we introduce a novel approach to deep learning. In the past there has been astonishingly little work on analyzing the internal structures of neural networks. While some work has gone into de-obfuscating the black boxes, such as TNN's, the re-use of internal logical structures remains unexplored.

Given a subtree component of a neural network, the next step is to figure out when to use it. The hope is that the contexts in which we find these components will hint at when the component should be used. By creating an index where each component maps to surrounding components (prioritizing more commonly connected components), we will make it possible for a search function to find the components it needs when it needs them.

Like in n-gram NLP models, context is a rather vague concept, and can vary based on *how much* context you are willing to encode. While in NLP models, context refers to surrounding words, here it refers to surrounding structures. Depending on limitations in computing power and storage, we may want to increase or decrease the amount of context the algorithm is allowed to encode.

Search.

Once we have an index, we can put the components to use. There are several ways we could go about this. The first is to search a neural network while it is training for structures that appear in our database. When a structure is found, the index is queried for relevant components. The index returns a list of potentially useful neural network patterns. A component is selected, and its weights are applied to the current neural network. The machine learning algorithm proceeds to run gradient decent as normal. If the changed weights don't change much further by the time the algorithm finishes, positive feedback is sent to the index, telling it to further prioritize that particular structure.

There are several problems with this, the foremost being that our database was not made from neural networks in training, but from fully pre-trained algorithms. Analogies based on half-baked concepts may not be very useful. Perhaps it would be more useful to train the network layer-by-layer, ensuring that we have a set of fully-developed components before querying the index with them. This, of course, introduces a slew of new problems, as it would require far more modification to existing machine learning algorithms.

The other main problem is what to do if a component doesn't fit in the neural network. The simple solution would be to ignore it and move on to the next one in the list. The complicated solution would be to modify the neural network to a certain extent to allow the inclusion of the pattern. In this project, we will probably stick with the simple solution.

Then there is the question of how often we want to use this tool. This will depend on how much time the algorithm saves and how much computing power it consumes. Once the system is up and running, we will experiment with the trade-offs. We would want to make full use of previously learned knowledge, but not depend on it too much. While many subproblems may be solved in other algorithms, many are not. So, an algorithm must find balance between using the knowledge of others, and learning for its self.

STS Problem

On the surface, it may seem that artificial intelligence gets plenty of attention in terms of publicity, funding, and general hype. However, AI is not a monolith. The sort of AI that has become so popular in recent years, deep learning, is but one sub-category of the AI field. Other sub-fields, such as evolutionary computation, planning, and general intelligence have received comparatively little attention. Fears of an oncoming "AI Winter" are constantly stirring, the fear that progress in AI will hit a wall. In this thesis I be approaching the topic from the perspective of the Sociology of Scientific

Knowledge (SSK), determine if the field of AI as a whole may be suffering from the paradigm of Deep Learning.

Importance

An AI Winter would be bad for everyone. As shown by past AI Winters, when innovation halts, businesses which were planning for growth find themselves unable to maintain it. In general, we would also be missing out on future technology if we devote all of our research time towards a dead end. If we can see a wall coming, however, we might be able to direct funding towards more long-term fruitful endeavors.

AI Winter

To start, we will delve into the past to investigate the social and economic factors behind previous AI winters. Multiple booms and busts in hype have occurred over the history of AI. Some of the first neural networks were being worked on in the fifties before being abandoned in favor of symbol-system AI's, until the concept was revived in the 80's. While the various revivals of neural networks can be attributed in-part to the creation of more-powerful computers, there were a number of social factors at play. One example is the infamous Lighthill Report, a report made for the UK Parliament in 1973 describing the failings of symbol-system AI, resulting in massive funding cuts to AI programs in the UK and across Europe (Lighthill 1973). There were counterarguments made, but the damage had been done, and the paradigm of symbol-system AI collapsed. By comparing and contrasting the AI field then and now, we could speculate whether or not a similar collapse will happen.

From there we will investigate the rise of the current Deep Learning hype. While neural networks are undeniably useful from a technological perspective, social and economic factors alike worked to both help and hinder their rise to dominance. The battles between "Good Old-Fashioned AI" (GOFAI) and "Connectionism" may seem like ancient history to today's data scientist, but many of the points made against neural networks by cognitive scientists in the 90's may still be valid today; namely, there was the argument that neural networks don't carry inherent meaning, and are hard to interpret.

The arguments were overshadowed by the practical success of neural networks, but never truly resolved.

The Deep Learning Paradigm

Deep learning refers to the class of functions which make use of multi-layered artificial neural networks (ANN's). An ANN can be treated as a "universal function approximator," (Fortuner 2017) meaning that a large enough neural network combined with the right set of internal values can approximate any function. This is useful in cases where one knows both the correct inputs and outputs of a function, but not how the function actually decides; the decision process can be approximated by slowly adjusting the internal values of the neural net until it accurately predicts the outputs given the inputs.

This effectiveness comes at the cost of time. Deep learning requires a lot of computing power to fully train. The more complex the problem, the longer it takes. So while it was first proposed in the late eighties, it wasn't until about a decade ago, when computing power rose to the task, that it became popular.

Effects on Sub-Fields

Sub-fields of AI such as Computer Vision, Speech Recognition and Natural Language Processing (NLP) seem to have benefited the most from Deep Learning. Most notably, the application of backpropagation to the convolutional filters used in Vision to create convolutional neural networks (CNN's) was a resounding success. It would be interesting to investigate, however, the topics being studied in these sub-fields before and after the adoption of deep learning methods, to see if any efforts were abandoned in the rush to use deep learning. Particularly NLP. Computer vision and speech recognition have something in common: they both involve bringing noisy data from a complex space down to a simpler space. Natural language processing also does this, but not to the same degree. Language processing requires not only recognition of patterns, but knowledge of the world in general: It is not enough to know a word is a noun; one must also have knowledge of the real-world object the noun refers to. So while modern text-prediction models like OpenAI’s GPT-2 can use deep learning to produce grammatically correct sentences, and can even mimic tone in some cases, most of the sentences are ultimately nonsense (Radford 2019). Perhaps focus on neural nets is holding the field back.

Modern Funding Efforts

With the history thoroughly investigated, we also need to look at the modern situation. What AI research, for deep learning and otherwise, is being funded and by whom? How much innovation has occurred in the past five years? Is there any sign of the deep learning hype slowing down? By identifying what artificial social factors are causing and sustaining current deep learning research, perhaps we can foresee potential problems.

Works Cited

Chi, Y., Muntz, R. R., Nijssen, S., & Kok, J. N. (2005). Frequent subtree mining–an overview. *Fundamenta Informaticae*, *66*(1-2), 161-198.

Fortuner, B. (2017, March 11). Can neural networks solve any problem? Retrieved from https://towardsdatascience.com/can-neural-networks-really-learn-any-function-65e106617fc6.

Lighthill, I. (1973). Artificial Intelligence: A General Survey. In *Artificial Intelligence: A Paper Symposium. London: Science Research Council*.

Kedar-Cabelli S. (1988) Analogy — From a Unified Perspective. In: Helman D.H. (eds) Analogical Reasoning. Synthese Library (Studies in Epistemology, Logic, Methodology, and Philosophy of Science), vol 197. Springer, Dordrecht

Radford, A. (2019, December 10). Better Language Models and Their Implications. Retrieved from https://openai.com/blog/better-language-models/.

Woolf, H. B. (1971). In *The Marriam-Webster Dictionary, Pocket* (pp. 433–439).