Assignment 1, WASP Software Engineering Course Module 2023

Oskar Nordenfors

August 2023

Introduction

In this essay, selected topics from the field of software engineering (SE) will be discussed in relation to my research project. I will begin by introducing my research project, which is a mathematics project, more specifically the mathematics of artificial neural networks. The project concerns the training dynamics of neural networks, i.e. how neural networks behave during training, specifically the dynamics of equivariant neural networks and augmented neural networks. By an augmented network we mean a network which is trained on augmented data, e.g., data where we have added rotated versions of images in the training data to the training data, in order to make the resulting model more robust. By an equivariant network we mean one whose architecture is equivariant by design, i.e. one where transforming the input is the same as transforming the output, e.g., rotating the input is the same as rotating the output. It is interesting to study such networks, since there are many problems where we encounter symmetries, such as rotational symmetries, which we want our model to incorporate (e.g., imagine that we wish to create a piece of software which can determine from the input of an image of a dog, which way the dog is facing. If the dog is rotated, the way it is facing, and hence our function's output, has to rotate along with it). The aim of the project is to study and compare the dynamics of such networks using a fairly recent tool in the literature, known as the Neural Tangent Kernel [1]. The Neural Tangent Kernel describes the training dynamics of neural networks in the infinite-width limit. First, we are looking at what sort of things we can say about augmented/equivariant networks in the infinite-width limit, that we could not say about them in the traditional setting. Then, we seek to expand our analysis to a broader class of networks. Finally, we hope to be able to say something concrete about the finite-width case, using what we have learned about the limit case, through the application of some measure concentration inequalities.

Concepts from the lectures

What I feel is the main idea from Robert's lectures is the notion that there is more to machine learning (ML) than simply creating better, more accurate models. E.g., if one is developing a product at a company, there can be many requirements that need to be met or certifications that need to be passed, there can be difficulties keeping the model updated as other processes at the company change, and there can even be difficulties determining exactly what kind of model should be developed given the companies goals. One way that this relates to my own research project is that given that the task we want to solve is one involving some symmetries we wish to exploit, equivariance is a property of huge importance. If we can understand more about equivariant architectures and how they relate to, and compare with, the method of using data augmentation (e.g., to induce equivariance), which is very often used in practice, this could have an impact on SE. Theoretical guarantees on stability of data augmentation methods to ensure equivariance could have an impact on the

SE side of things by reducing the amount of testing that needs to be done, which could free up valuable resources if the models are being applied to some critical area, where much testing has to be done. Note that this connection to my research is tenuous at best, as well as being entirely speculative in nature. This topic does relate to the second concept I took away from Robert's lectures, which is the fact that testing is in general a big challenge in SE and perhaps an even bigger challenge in ML, where the models we work with have large amounts of parameters which can be hard to understand in the same way that one can understand traditional software code. This could turn out to be a big challenge in my research project as we will want to create equivariant models and augmented models to compare against each other in empirical experiments. I know by cultural osmosis that many researchers are still finding bugs in their code up until the day their papers are published, and sometimes even after. So testing these models to make sure they meet our specifications could become a hurdle we will have to overcome. But there was some good discussion about testing during both Robert's and Felix's lectures, which has me at least thinking about the potential issues I might face in my own project. It is a bit less clear what, if anything, from the other lectures I can relate to my own research, but I will try to relate to a couple of topics from Linda's lecture. Before Linda's lecture, I was not aware of how far along the code helper bots were, and this is now something I am looking into for when I will be implementing my models in e.g. Python at the later stages of the research project. I will of course not be writing any huge amounts of code, but these kinds of "autocomplete"-like bots could still be quite useful, I think. The other concept from Linda's lecture that I have been thinking about afterwards is the SPACE framework, or in general any kind of productivity framework. I feel that this should be of use in a research project with several PhD-students and supervisors, to make sure that everyone is staying healthy and happy and are able to perform their best. It cannot be that this is only a useful concept in a company context; it should also be useful in a research context. We still have something like a pipeline that we are pushing a product through. In our case the product is a research paper with many components, both theoretical and experimental. So the theoretical component could be seen as a sort of requirements engineering that we then have to match with our implementation. So, the set-up is quite similar as in a software engineering context. Of course there are differences as well, but I am focussing on the similarities here. Of course performance and activity are of importance still in this context. The same goes for communication and collaboration, which is important as part of a larger research group. It is also the case that one wants to be efficient and have a good workflow. This covers all of SPACE, but one can make similar mappings for other frameworks.

Discussion of selected topics

This part of the assignment was incredibly difficult to write for me, since my research really has no relation to software engineering. Which makes discussing how my research topic relates to these different topics from ML engineering rather

difficult. Keep this in mind when reading the following discussion. The first topic I have chosen is quality assurance for ML (QA4ML) and the paper I have looked at to inform my writing is [2]. My understanding of QA4ML is that there are many challenges which arise from the black-box nature of ML systems. Traditional software is hand-designed by engineers through an intelligent process, which is at least in principle not a black-box and can be understood by careful consideration, whereas ML systems are full black-box, created from data (which is often hard to understand itself) through a process which is to a large extent unintelligent or automated (fancy statistical methods), and there is often really no way to get a handle on the many parameters and their obscured meanings. Of course this affects work with quality assurance (QA), since it is quite difficult to assure the quality of a product that one does not even in principle have a real chance of understanding. I am aware, however, that even traditional software systems can become basically a black-box as their size and complexity spiral out of control. But, I think it is fair to say that the problems are much bigger in ML engineering. I would say that my research in mathematics falls somewhere in the realm of explainable ML (XML), though to what extent is unclear since it is still very early days. The field of XML is, I think, certainly connected to the topic of QA4ML, since through designing more easily explainable models we can ensure that QA can proceed more smoothly and with greater ease. If one has a better understanding of a system, then one does not need to run as many tests, and can run better and more relevant tests, etc. This can be even more relevant if the ML model is a part of some larger product, with customers who are expecting certain requirements to be met, since we do not want one single part of the system to make our testing too difficult and costly. My project also concerns data augmentation, which is a regular part of many ML procedures in various applications. Here is another opportunity to have an impact on QA4ML, since if we can better understand data augmentation, then one can more easily assure the quality of the data, which is, of course, an important part of QA4ML (garbage in, garbage out). The second topic I have chosen is requirements for ML engineering (RE4ML) and the paper I have chosen to inform my writing is [3]. My understanding of RE4ML is that there are many aspects to requirements engineering in both SE and ML. There are requirements from the stakeholders that need to be discussed, there is analysing and documenting them, and there is checking that documented requirements match what the stakeholders needed to begin with. My project being very focussed on the design of ML models relates to this in a fairly narrow sense, since not all requirements are made of the model, but can be made of other parts of the whole product pipeline. E.g., there could be requirements made on the data that is being used to train the model. Here, I chose this particular example since it is somewhat related to the part of my research that deals with data augmentation. But there could be other requirements on things other than the model or the data, which would be outside the scope of possible relations to my own research. However, for model-based requirements, a deeper understanding of equivariance (invariance) and stability of equivariance (invariance) for augmented networks could help with planning and meeting requirements. E.g., we want our model to not discriminate

based on gender, i.e., we want the model to be invariant to the gender of some person being used as input to the model. Or maybe we want a medical treatment to vary with the gender of some person being used as the input, since the wrong treatment for the wrong gender may be disastrous (e.g., wrong dosage of medicine). But again, this is still a fairly narrow sense of requirements engineering, since it only bears on the model/data. However, as far as my own research area is concerned, the main opportunities for meaningful research contributions on this topic, I believe come from the stability of equivariance (invariance) in augmented networks, since equivariance (invariance) is of great importance in many real-world applications e.g., computer vision, fairness for medical solutions, etc. So, this covers the connections between the two chosen topics to my area of interest. Some of the connections here are quite vague, so I think it is important to take this with a grain of salt, and to not overestimate the potential impact of my research on ML engineering practices, even though data augmentation is a part of many ML pipelines. After all this is not the focus of my work. I am merely laying the theoretical groundwork for further research, and am not really focussed towards applications, despite appearances.

Future trends in ML/AI engineering

I am not an oracle, so I can only speculate about the future, which usually is an endeavour that ends up with one looking like a fool. On the ML for SE side of things I think that the current trends with the largest tech companies chasing bigger and better large language models (LLMs) will continue for the foreseeable future. I do not think that the recent controversies, and media storm surrounding them, will do much to change that. I think that this will probably have a big impact on the workflow of software engineers in the coming years, as it is already starting to have, considering the effectiveness of using, e.g., gpt-4 to help with coding. However, unless models appear which are much better at mathematics (which could happen) I do not see it having a big impact on my own research/field. On the SE for ML side of things, I think that as more and more companies and government institutions start to implement ML models for performing various tasks, there will be many challenges to overcome and problems to solve. Especially concerning testing procedures and explainable ML. Here, I would like to think that research like my own has the potential to play some small part, since we are concerned with providing a theoretical understanding of two ways to create simpler, and thus more explainable, models; namely, models with equivariant architectures and models trained to be equivariant through data augmentation. Although, any impact would be necessarily limited to applications where there is some group symmetry at work. I think for AI, mathematics will be playing keep-up with the state-of-the-art, as it is currently doing, and us mathematicians will slowly but surely be able to explain more and more of the phenomena that people discover in practice, while not really having a massive influence on best practices in the field of ML engineering.
This covers all of the questions in the assignment.

Bibliography

[1] A. Jacot, F. Gabriel, and C. Hongler. Neural tangent kernel: Convergence and generalization in neural networks. *Advances in Neural Information Processing Systems*, 31:8571–8580, 2018.

[2] J. M. Zhang, M. Harman, L. Ma, and Y. Liu. Machine learning testing: Survey, landscapes and horizons. *IEEE Transactions on Software Engineering*, 48(1):1–36, 2022.

[3] S. Nalchigar, E. Yu, and K. Keshavjee. Modeling machine learning requirements from three perspectives: a case report from the healthcare domain. *Requirements engineering*, 26(2):237–254, 2021.