On the Nature of AI Code Copilots

Stuart Fitzpatrick

School of Computing, Data and Mathematical Sciences - Western Sydney University

ABSTRACT

The recent release of the GitHub Copilot an ‘AI pair programmer’ trained on the billions of lines of publicly viewable code has brought to the forefront discussion on the very nature of Machine Learning and Artificial Intelligence systems. This paper specifically addresses the following issues: Whether they constitute a compiled form of the training data or if they are more akin to a computer program’s source code, and whether they are in violation of copyright.

S.Fitzpatrick2@westernsydney.edu.au

1. Introduction

On the 29th of June, GitHub announced the imminent release of GitHub Copilot a so-called ‘AI pair programmer’, developed in collaboration with OpenAI (Friedman, 2021). This resulted in two predominate types of reactions, the first was a wave of amusement as individuals with access to early versions of the program reported issues such as; attributing an incorrect license to well known pieces of code (Ronacher, 2021), generating an ‘about me’ page with another individual’s information (Peacock, 2021), and using floating point values to represent currency (Copilot, 2021).

The second and more pressing reaction was the question of how does this relate to software licensing? Given examples such as (Ronacher, 2021) it is readily apparent that GitHub Copilot is capable of returning, verbatim, already extant code (although it does attempt to synthesise novel code based on its training data). This immediately raises the issue, what happens when that code (such as the previous example) is licensed under a copyleft license such as the GPL or AGPL? How is the matter of copyright in this instance resolved?

The resulting issues that need to be considered are outlined here:

1. Do AI pair programmers (and ML systems generally) constitute a ‘compiled’ form of the training data, similar to an executable file generated by a compiler?
2. If they do not, then are they closer to the source code of a conventional program that can be modified by doing further training?
3. How does this affect the resulting copyright of the code suggested by the copilot?

2. What exactly is an Artificial Intelligence system anyway?

The idea behind artificial intelligence is, fundamentally, quite a simple one. It is a system that takes information about its environment in some form (such as image, video and text data) and performs, without human intervention, some series of steps that maximises the chance of achieving some goal (Poole, 1998). This goal could be making video recommendations based on previously viewed content, or producing text that constitutes computer code and comments (as in the case of AI copilots).

Artificial intelligence systems that learn by the use of data or ‘experience’ are perhaps the most common form of artificial intelligence and are referred to as Machine Learning (Mitchell, 1997). Machine learning algorithms can be broadly divided into two categories; Feature Engineering, where what parts of the data are important to determining the outcome are known a priori, and Deep Learning where these features are not known a priori and are inferred as part of the learning process. GitHub Copilot is a system of this second type.

3. How is Machine Learning Implemented?

Regardless of the type of algorithm that constitutes a machine learning model (model here meaning specific implementation of a specific algorithm), they all fundamentally work in the same overarching way. Creating a machine learning model consists of four distinct phases.

1. Training, where the algorithm attempts to return correct outputs given some inputs.
2. Validation, where the algorithm is tested on some combination of new and reordered inputs to ensure that any bias towards the training data is minimised.
3. Testing, where the algorithm has its final performance evaluated on the task that is of interest (note this does not have to be the same task the model was trained on).
4. Deployment, where the algorithm is placed in some piece of software that provides an interface for providing input and returning output.

For deep learning models, because they infer features during the training process, particularly large and computationally demanding ones can go through a paired process known as pre-training and fine-tuning. In this process the model is trained on inputs from an entire class of tasks, such as answering questions, obtaining moderate performance in all possible varieties of that task, then as needed, it is trained again on demand for specific subsets of the class of task, such as answering customer support questions. GitHub Copilot is an example of a fine-tuned model, specifically it is a fine-tuned version of OpenAI's Generative Pretrained Transformer - 3 (GPT-3).

4. Compiled or Source, Where do Artificial Intelligence Systems Fit?

Consider that when a program is compiled from source into a binary file, the compiler is taking some code written in a programming language, and it returns as output the equivalent instructions in the target platforms assembly language (typically this is x86_64 or ARM, but others exist). Given that apart from need to tell the compiler what the target platform to compile is, the entire process can be completed without the need for human intervention.

Now, recalling that a specific trained result of a machine learning system takes some input and returns some specific output, it is clear to see that there are more than superficial analogies between the two that can be made. Indeed it could easily be said that, because a compiler is written by programmers who are (mostly) humans, it is not an unreasonable to stretch to refer to a compiler (and in a similar vein, interpreters) as Natural Intelligence Systems and Natural Learning Systems.

Furthermore, it is commonly accepted, amongst machine learning practitioners and data scientists, that fine-tuned versions of a specific deep learning model, is not some new and distinct program, separate from the original, it is (a form of) the original, in the same vein as the Ship of Theseus. Explicitly using GitHub Copilot as an example, as it itself is an example of a fine-tuned (GPT-3) model, to machine learning practitioners and data scientists, GitHub Copilot is merely the trade name of a specific implementation of GPT-3.

Turning our attention back to compilers, if the source code of, for example Grep is taken by a compiler and compiled into two different assembly languages, say x86_64 and ARM for argument's sake, are both of those programs still Grep? Or are both programs something distinct from both each other and from their source code? They are both still clearly Grep, and it is by that reasoning a program such as GitHub Copilot is still GPT-3.

Finally, consider this that if the training data of a machine learning model is altered, the model at the end of training is also altered (specifically it will give slightly different outputs for the same input). This alteration however, also does not change what the algorithm itself is (or is doing). Likewise when the source code for a program is changed, the resultant output (i.e. the executable) is also changed, but again this change does nothing to affect what the compiler itself is, or what it does, it merely gives slightly different output for some input. Being more explicit, the source code for grep is still the source code for grep, even if, for instance, all the variable names in the code are changed to be their first three letters, and so is a deep learning model still the same deep learning model even if its training data is changed (or the training process resumed later on, in the case of fine-tuning).

Therefore it can be plainly and simply stated that an artificial intelligence system such as a neural network, like GitHub Copilot can be considered, not akin to the executable generated by a compiler, but rather are significantly closer to being source code than a simple executable file.

5. What About Copyright?

It has been shown that GitHub Copilot is capable of regurgitating training data verbatim and returning it as output as evidenced by (Ronacher, 2021), this is a well known phenomena in machine learning and so will have likely been expected, at some level by the developers of by GPT-3 proper and the GitHub Copilot team. In the fields of machine learning and data science, it is typically not considered copyright infringement for a model to return an input as an output.

Strictly speaking in a legalistic sense however, all that is required for a copyright violation is for the unauthorised reproduction of the work in a manner that does not fall into the concepts of ‘fair use’ or ‘fair dealing’, so it needs to be determined if the a machine learning algorithm returning input as output falls into either of those categories.

5.1. Fair Dealings & Fair Use

Typically, most machine learning practitioners and data scientists are engaged in active research, where the concept of fair dealings and fair use, typically grant an exception. GitHub Copilot on the other is not the product of such mere academic interest, but rather was designed from the top down with the intention of being a commercial product. This commercial nature almost certainly infringes the copyright of the various instances of training data, if this factor alone is considered.

5.2. Authorisation?

The question then becomes, if GitHub Copilot cannot rely upon the concept of fair dealing and fair use, then it must have surely obtained the permission of all the copyright holders? This, is where the case becomes murkier. According to the GitHub Terms of Service, by storing your code remotely using their service you (the end user) grant GitHub certain rights including ‘‘This license includes the right to do things like copy it to our database and make backups; show it to you and other users; parse it into a search index or otherwise analyze it on our servers; share it with other users; and perform it, in case Your Content is something like music or video." (GitHub, 2021) The ToS however then further states that ‘‘This license does not grant GitHub the right to sell Your Content. It also does not grant GitHub the right to otherwise distribute or use Your Content outside of our provision of the Service" (GitHub, 2021). The question then remains, does the small rate at which GitHub Copilot returns, verbatim, copyright infringement constitutes a reasonable attempt to use the publicly hosted code on GitHub to improve the service? Although legally it would be ultracrepidarian for this author to make a judgement in that area, in terms of philosophical reasoning however, this is clearly above what could and should, by most people be considered reasonable scope given that GitHub Copilot is a commercial code plugin for Integrated Development Environments (in particular the proprietary VSCode), that has nothing to do with the GitHub service apart from sharing the branding and perhaps some of the development team. By the given metrics, GitHub Copilot is quite easily within the bounds of what most would consider to be copyright infringement, and is therefore in breach of both copyleft licenses such as the GPL and proprietary licenses that withhold the permission to view or otherwise disseminate the source code of a program.

6. Conclusion

To conclude, given the properties of how AI code copilots function, when needing to determine whether they are closer to source code or executable files, they are much closer to source code, than to executable files. Further given that the verbatim return of inputs does not fall under fair use, nor does it obtain the authorisation of the copyright holders to use their code in such a manner and are as such in violation of both copyleft and proprietary software licenses.

References

Friedman, 2021.
Nat Friedman, *Introducing GitHub Copilot: your AI pair programmer* (Jun 2021). https://github.blog/2021-06-29-introducing-github-copilot-ai-pair-programmer/.

Ronacher, 2021.
Armin Ronacher, *I don’t want to say anything but that’s not the right license Mr Copilot.* (Jul 2021). https://twitter.com/mitsuhiko/status/1410886329924194309.

Peacock, 2021.
Kyle Peacock, *Congrats @davidcelis, you get a shout out if #GitHubCopilot tries to generate an "About me page"* (Jul 2021). https://twitter.com/kylpeacock/status/1410749018183933952.

Copilot, 2021.
GitHub Copilot, *GitHub Copilot Official Preview* (Jun 2021). https://copilot.github.com/, parse_expenses.py.

Poole, 1998.
David Poole, Alan Mackworth, and Randy Goebel, *Computational Intelligence: A Logical Approach*, Oxford University Press (1998).

Mitchell, 1997.
Tom Mitchell, *Machine learning,* McGraw-hill New York (1997).

GitHub, 2021.
GitHub, *GitHub Terms of Service* (2021).