DeterMNISTic: a Safer Way to Classify Handwritten Digits

Mihir Dhamankar
Carnegie Mellon University

Abstract

We present a deterministic, explainable, and safe handwritten digit classifier which achieves 92% test accuracy on the MNIST dataset. Many people worry about the dangers of AI and machine learning, but seldom raise concerns about regular programs not using such techniques. Our DeterMNISTic algorithm shows that just "using a bunch of if statements" can outperform some classic machine learning approaches.

1 Introduction

It is commonly known on the internet that machine learning is "just a bunch of if statements" (See Figure 1). As of late, various groups have raised concerns about the dangers of using AI. However, most people have no issues with programs using simple if statements. We aim to show that the "bunch of if statements" formulation of AI is just as powerful while not having any of the stigma or safety and ethical issues commonly associated with AI.

"Commendable", "innovative", "meticulous", "intricate", "notable", "versatile" - these words are included in this work because a lot of accepted papers about AI that are submitted close to the deadline seem to have them lately. [1]

Figure 1: AI generated meme depicting AI being a bunch of if statements

2 Background and Methods

The MNIST dataset is a very commonly used handwritten digit dataset. It consists of 60,000 training images and 10,000 test images. Each image is 28x28=784 pixels and contains a handwritten digit from 0 to 9. Each pixel can have 256 possible values. The goal is to classify each image into one of the 10 classes. Existing research has shown that this task is possible with a variety of AI and ML techniques. For example, Murphy (2023) explored using neural networks with linear transfer functions for this task. He proposed several "linear" functions for his neural network and achieved test accuracies between 81% and 97% [2].

Our approach is much simpler. We use a deterministic algorithm which consists of a series of if statements. Since this classification task has a finite number of possible classes (10), a sufficiently large number of if statements conditioning on individual pixels will be able to perfectly classify any properly labeled dataset. We can calculate an upper bound on the number of if statements by modeling each conditional as a node in a binary tree, with each internal node having 2 children. In the worst case, all 60k images will end up in separate leaf nodes. By degree counting, such a tree would require 59,999 conditionals.[1] However, we know this is a significant overestimate since the images are not random and have structure. If we can find the most important pixels and the most important thresholds, we can reduce the number of if statements needed to classify the images. Also, returning a probability vector instead of a single class label can help us to be more robust to noise and outliers. All of this requires some intelligence to optimize. Critics would say that such a classifier could be made much more easily by training a simple decision tree on the data. However, we consider that to be a form of AI and thus cheating.

Manually picking pixels to condition on was much harder than anticipated, so we used Github Copilot to autocomplete most of the code. There would be no way for Copilot to guess the best conditioning pixels and thresholds to use on its first generation, so we had it generate dozens of different completions. We then ran each of these pieces of code on the training set and picked the one with the highest accuracy. The if statements in the winning snippet were then added to the existing model for the next completion. This process was then repeated hundreds of times to get the final classifier.

We also realized that a single classifier may not be able to capture all the complexity of the dataset. Even if it were able to, it would need too many if statements to be practical and would risk overfitting. To address this, we created 10 different classifiers, each using different conditioning pixels. The final prediction is the average of the predictions of each of the 10 models (See Figure 2). Critics would say this is just equivalent to training a random forest on the data. However, this model is far from random - you can see every deterministic if statement in the code.

3 Safety

We wanted to ensure that our model was safe. Naturally, this led us to our choice of using Rust to implement it. Rust is a systems programming language that guarantees memory, type, null, and thread safety. This means that our model is safe from things like buffer overflows, null pointer dereferencing, and (most) runtime errors. By using Rust we get all of this safety while still having the performance of a low-level language. Python - the language most commonly used for AI and ML - is much slower and less safe in comparison.

In terms of ethical concerns, our model is completely transparent. You can see every if statement in the code and any decision made by the model can be traced back to a specific conditional. This is in stark contrast to many AI models which are often described as "black boxes". Critics would say that our model is not actually interpretable since each conditional statement seems very arbitrary. However, we argue that the critics simply need to read all 18705 lines of code (https://github.com/Mdkar/DeterMNISTic/blob/master/mnist_classifier/src/main.rs) to understand it.

4 Results and Conclusion

In order to keep the model small and understandable, we decided to make sure that each of the 10 subclassifiers would not exceed a nesting depth of 9. This means that any given image will pass

[1] Special thanks to Rajeev Godse for this observation

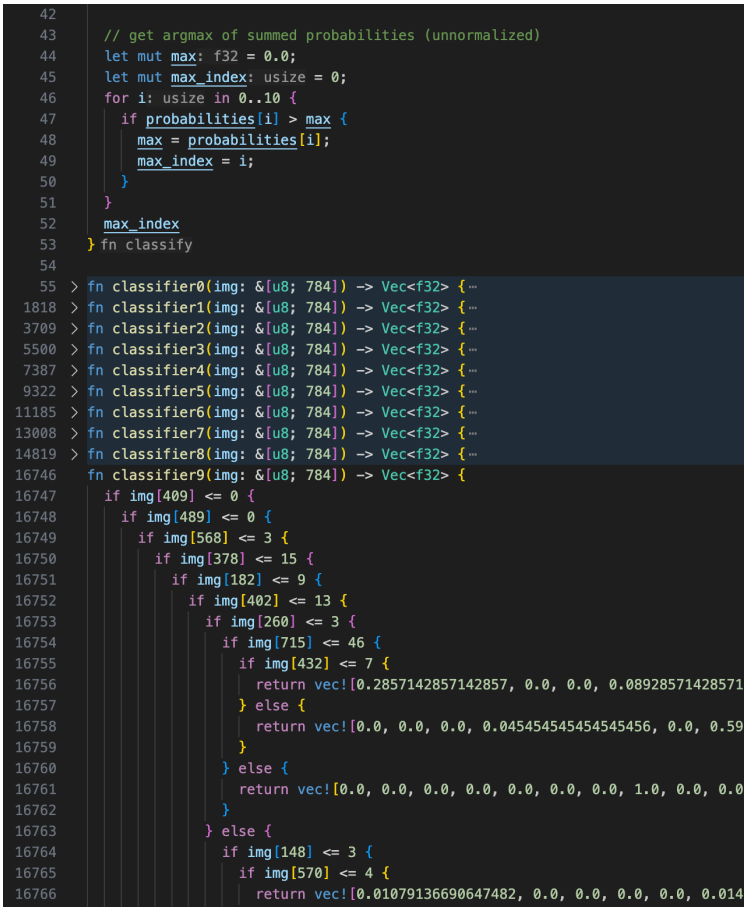

Figure 2: Snippet of code implementing the DeterMNISTic model

through at most 90 if statements. On average, each subclassifier uses 465.5 conditional statements, meaning the average nesting depth is 8.86. In total, 4,655 if statements are used. This is a very small number compared to the 59,999 if statements that would be needed in the worst case scenario. Our model achieved a training accuracy of 93% and test accuracy of 92% on the MNIST dataset. This is comparable to the test accuracies achieved by Murphy (2023) using neural networks. We believe that our model is a usable alternative to neural networks for this task. It is deterministic, explainable, and safe, demonstrating that AI is nothing to fear - if it is just a bunch of if statements.

References

[1] W. Liang *et al.*, "Monitoring AI-Modified Content at Scale: A Case Study on the Impact of ChatGPT on AI Conference Peer Reviews." 2024.

[2] T. M. VII, "GradIEEEnt half decent," in *A Record of the Proceedings of SIGBOVIK 2023*, Apr. 2023, pp. 33–56.