

Connect 4 With Machine Learning

<Date>

Table of Contents

[Analysis 3](#_Toc52958804)

[Introduction 3](#_Toc52958805)

[Identification of Programming Language 3](#_Toc52958806)

[Numbered Objectives 4](#_Toc52958807)

[Documented Design 5](#_Toc52958808)

[Technical Solution 6](#_Toc52958809)

[Testing 7](#_Toc52958810)

[Evaluation 8](#_Toc52958811)

# Analysis

## Introduction

The problem area that I am interested in is Machine Learning which is a large topic in computer science research. I am planning on creating a machine learning algorithm which will train a neural network to act like a human at playing the board game, Connect 4. The main reason I chose this was because I watched a video on machine learning[[1]](#footnote-1) and was instantly mesmerized. I decided that I wanted to learn how to make something similar and chose Connect 4 as the game because it is quite simple and can demonstrate a learning AI quite well.

Machine learning algorithms have been around since the 1950s but has only recently taken off due to the rise of the internet and masses of data being transferred daily. Companies such as Google and Facebook are using machine learning algorithms daily to provide a better user experience, better than any human could do on their own. Many tools have been made to allow machine learning to be used by many people without much background in the topic itself. One of my inspirations is Google’s DeepMind[[2]](#footnote-2) AI which taught itself to beat a world-champion at the Chinese game of Go[[3]](#footnote-3).

Machine learning is based on mathematical ideas to adjust parameters such that a cost function is minimized. In my case of Connect 4, the cost function will be how different the AI’s choice is to the human’s choice so that when training, the AI will minimize this and start imitating the human better. I will derive all mathematics behind the algorithm as it is quite complex. The implementation is simple but, can get quite complex for efficient code due to a few tricks in organising the data in matrices and using matrix math to do calculations.

I am planning on only using a single Machine Learning algorithm for my final product as the number of tweaks to parameters I can make to the algorithm is massive. I should be able to adjust the algorithm so that I can get the most performance out of the single algorithm. If I finish the algorithm and realize that I there aren’t enough parameters to tweak, I can add another algorithm but this will add a lot of complexity and take a lot of time as a large rework will have to be made. This is because I would have to allow a modular kind of program where the number of algorithms available is not hardcoded, but more can be added at any time.

## Identification of Programming Language

Research about programming languages came from Ignite Digital[[4]](#footnote-4) and Raygun[[5]](#footnote-5).

* **Python** – Python is a very user-friendly language with many libraries that help development. It does supports Object Oriented Programming and has a lot of Quality of Life features built into it however, it is not the most performant language when coding at a basic level. I have a small amount of experience in Python but will have to research how it handles object-oriented programming.
* **Java** – Java, like Python, has an abundance of open-source libraries useful for certain applications (including Machine Learning). Again, it is not the most performant language but, has much more control over lower level ideas than Python does. I do not have much experience at all in Java, but the syntax is extremely like C/C++ meaning it will not take much to learn.
* **C/C++** – Both C and C++ are extremely low-level languages allowing for deep control of hardware to gain large performance boosts. C++ comes with its own libraries (STL[[6]](#footnote-6) – Standard Template Library) which are highly performant and almost never ignored for another library. Only C++ is Object Oriented out of the two meaning C will require a bit more work for certain areas. I am extremely familiar with C++ as I use this language for almost all my projects.
* **C#** – C# is an exceedingly popular language for development as it allows for low-level control like in C but also is much more user friendly like Python. C# is object oriented which is useful and certain programs such as Unity use C# as their language of choice meaning Graphics with C# is a lot easier than with other languages. I have almost no experience with C#, but the syntax is similar to C/C++ with simple pieces of code but changes a lot with function definitions and similar ideas.
* **Visual Basic** – Visual Basic is a simple and pseudocode like language meaning it is very user friendly with how it works. However, this causes some problems when it comes to consistency as some differences from other languages become apparent when using arrays as many times Visual Basic handles these differently and *off-by-one[[7]](#footnote-7)* errors occur are quite common. Visual Basic is the language we have been learning at School since year 10 meaning, I am quite familiar with it but due to its awkward Object-Oriented Programming it will require some learning.

In conclusion, after researching different programming languages, I have decided to use C++. This is because of its low-level control and great Object-Oriented Programming. The only problem with C++ is the use of Graphical Libraries, this is because as C++ is a language with a lot of control, the libraries it comes with also contain a lot of complexity meaning it is difficult to use them to do simple things. I would have chosen Python due to its user-friendly syntax however, I do not have a good enough understanding of how the Python compiler works in terms of multiple files to be able to create a large project with it.

## Numbered Objectives

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | Main Objective | Explanation | Success Criteria | | Sub-Success Criteria | |
|  |  |  |  |  |  |  |

# Documented Design

# Technical Solution

# Testing

# Evaluation

1. <https://youtu.be/gn4nRCC9TwQ> [↑](#footnote-ref-1)
2. [https://deepmind.com](.git) [↑](#footnote-ref-2)
3. <https://en.wikipedia.org/wiki/Go_(game)> [↑](#footnote-ref-3)
4. <https://www.ignite.digital/10-best-programming-languages-to-learn-in-2020/> [↑](#footnote-ref-4)
5. <https://raygun.com/blog/programming-languages/> [↑](#footnote-ref-5)
6. <https://en.wikipedia.org/wiki/Standard_Template_Library> [↑](#footnote-ref-6)
7. <https://en.wikipedia.org/wiki/Off-by-one_error> [↑](#footnote-ref-7)