

# Sick Arithmetic

Zachary Frank, Cedric, Emyl

October 31, 2019

## Abstract

With this project we will create an AI that will be able to perform basic mathematical computations. Users will be able to give the AI equations and will be able to evaluate the output of the AI. This way the AI will learn from its mistakes and will become better with every equation that will be given. Users will also be able to generate a report detailing the intelligence and progress of the AI over each iteration.

## 1 Role Assignment

Roles	Name	Task description
User	Zachary Frank	Provide feedback on updates to the project.
Customer	Zachary Frank	Give the software developer and development manager requirements based on user feedback and needs.
Software developer	Emyl	Lead the development and make software design decisions.
Development manager	Cédric	Project management and managing customer requirements.

## 2 Introduction

We wanted to combine AI and mathematics by teaching one to perform mathematical computations. It is our hope to create an application that will help people learn how an AI is trained using mathematical equations. This will give a user the ability to train their own AI from scratch and learn what methods are most effective. To do this, the client needs a friendly user interface that will allow them to communicate mathematical equations as input to the AI and receive feedback. The client also needs a way for users to evaluate the intelligence and progress of the AI they have been training.

### Similar software:

During our research we found that multiple attempts at teaching AI math has been done, for example by using the neural network api Keras. We also found research in the field of psychology where the neural network's learning of math is compared to that of a child's<sup>1</sup>.

The biggest project we found was DeepMind's attempt at teaching an AI to pass a high-school math exam, by teaching it arithmetic, calculus and algebra. Their AI scored 14 out of 40 on the exam. While the scope of this project is bigger than ours, it serves as a great inspiration!<sup>2</sup>

---

<sup>1</sup><http://web.stanford.edu/~kmickey/pdf/MickeyMcClelland2014.pdf>

<sup>2</sup><https://www.zdnet.com/article/ai-aint-no-a-student-deepmind-flunks-high-school-math/> & <https://arxiv.org/abs/1904.01557>

## 3 Requirements

### *Recognize Mathematical Syntax*

#### *a. Equals*

Learn the association between two numbers with an equal sign. Evaluate whether a statement is true or false, for example  $20 = 20$  is true,  $3 = 2$  is false. Also answer questions such as  $4 = ?$  and return 4.

#### *b. Addition*

Learn how to add the value of numbers on either side of a plus sign and find their sum.

#### *c. Subtraction*

Learn how to subtract the value of one number from another and find their difference.

#### *d. Combine Operations*

Combine several operations together such as  $5 + 5 = 10$  and evaluate if the statement is true or false.

#### *e. Negative Numbers (Optional)*

Perform the above operations with negative numbers.

#### *f. Multiplication (Optional)*

Learn how to multiply the value of numbers on either side of a multiplication sign and find their product.

#### *g. Division (Optional)*

Learn how to divide the value of numbers on either side of a division sign and find their quotient.

### *User Interface*

#### *a. Data input and output*

Create a user interface that has an input field for the equation and an output field where the AI can give the answer to an equation.

#### *b. User guided evaluation*

Contain a panel where the user can begin an evaluation for the AI. User should be able to create testing questions as well as complex testing structures (a course of many evaluations for example).

### *Evaluate training progress*

#### *a. Tests*

Formulate test scripts to query the AI in order to determine the effectiveness of training. These questions and tests should vary in difficulty and type in order to discover the most effective testing methods.

#### *b. Database*

Save the results of evaluations in a database. Data models should track students using an id number and aggregate multiple forms of data such as grades, grade schemas, and assignments.

### *Recognize hand drawn formulas (Optional)*

Use image recognition to provide another method of providing input.

## 4 Development Environment

### *Platform Choice*

This will run in a web environment so that it can be accessed by many users through a web browser. This will also relieve the heavy processing loads from the client machine and put them on the web server.

### *Languages*

The web server will be built using Python and the database server will be built using Java. This allows for different members to use languages that they are comfortable with.

### *Cost Estimation*

Jetbrains All Products Pack \$64.90 / user / month  
3 users, 2 months

\$389.40

MySQL Standard Edition

\$2,000.00

Amazon Web Services - EC2 t3a.xlarge

\$71.84 / month after first year

### *Development Environment Information*

Django 2.2.6

Python 3.6

Java 13

Maven 3.6.1

Spring Boot 2.1.9

MySQL 8.0.17

Apache 2.4.41

Ubuntu 19.10

Jetbrains PyCharm 2019.2.3

Jetbrains IntelliJ IDEA 2019.2.3

Keras 2.3.0

### *Commercial Cloud Platform*

With enough resources, the service will be hosted on an Amazon Web Services server. This will host the web server and database server components.

### *Software in use*

DeepMind has done research into neural networks ability for mathematical reasoning and processing. This includes simple arithmetic like our project, but also higher-level math such as probability and algebraic generalisation<sup>3</sup>.

<sup>3</sup><https://arxiv.org/abs/1904.01557>

## 5 Specification

### *Recognize Mathematical Syntax*

Mathematical equations will be represented as tree structures. These will be produced via lexical and syntax analysis much in the same way a compiler parses a program.

The tree will be used as training data (assuming the equation represented by the tree is correct) for a modular neural network. The network will split inputs to evaluate equations recursively.

Each operator should have a node that activates via sigmoid function by recognizing operators using ASCII decimal representations for input.

These calculations will be performed on the server using input received through REST API requests made from a client browser. The server will use Python Keras to train the network model and answer API requests using the trained model.

### *User Interface*

Using a web browser, the graphical user interface will seamlessly interact with backend services using Python Django. Everything the user needs will be accessible from a single page including training and testing the AI. Requests will be made asynchronously to a REST API in JSON format.

The user will be able to send equations to the AI through an input field. The input field will be created using the premade functionality that Django

provides. The input will be handled by Django and it will send the output from the AI to the user. After the user receives the output the user can tell the AI if the output was right or wrong using two buttons. One button for false if the output is wrong and one button for true if the output was correct.

### *Evaluate Training Progress*

#### *a. Tests*

By using prepared scripts, the user can evaluate the AI on its success/learning rate. These grades will be stored in a database for future reference on learning progress. There will be multiple tests that focus on different types of questions that test the AI on different operators and equations. The tests will contain multiple questions.

#### *b. Database*

The results of the aforementioned tests will be stored in a database. The database will track the result of each test the AI takes and its overall grade. The database will store assignments, grades linked to an assignment and a student(AI model), and grade schemas. Grade schemas will be structures that store many assignments that can be used to evaluate an AI across multiple topics.

Refer to Appendix A

### *Recognize hand-drawn formulas (optional)*

The user will have an additional way of inputting equations by writing by hand in the designated box in the user interface. The AI will use image recognition to recognize this input and then solve it.

## Appendix A

