

Plan:

Project Description:

The goal of our group's project is to create an app that would take a picture of a handwritten math equation and solve it. An easily accessible and convenient app would benefit math education around the world as students would be able to get instant feedback on their work.

For current day equation solvers, it can be tricky to type in various forms of equations, and systems such as LaTeX are not intuitive for typing equations. Therefore, having an app which could snap a picture of handwritten work would improve student efficiency, so using handwriting and image recognition could be beneficial for a student workflow.

Looking at this project big picture, there are a couple main goals we need to complete:

- Training and testing a model from a dataset to recognize handwritten equations
- Serializing the model and implementing it into some sort of backend
- Creating some frontend which can take a picture and pass it to the model
 - Ideally it would return the equation solved, but at this stage we might just focus on making sure it actually interprets the equation correctly

Sprint 0:

- Decide on a topic to work on
- MVP Document
- Plan.MD

Sprint 1 (2/25 – 3/11)

- Get familiar with TechStack
 - Complete tutorials on python, numpy, pandas
 - Get familiar with tensorflow, keras
 - put any research into the research folder
- Choose a dataset (for math handwriting recognition) to work with
 - Math and basic symbols: <https://www.kaggle.com/clarencezhao/handwritten-math-symbol-dataset>
 - This one might be a bit trickier: <https://www.kaggle.com/aidapearson/ocr-data>

Sprint 2 (3/12 – 3/25)

- Cleaning up and preprocessing dataset (if needed)
 - Creating training and testing datasets for validation
- Training and Testing a model using tensorflow
 - Attempt various classification models and see which has best accuracy
 - Some helpful tutorials to follow:

- <https://www.digitalocean.com/community/tutorials/how-to-build-a-neural-network-to-recognize-handwritten-digits-with-tensorflow>
- <https://machinelearningmastery.com/how-to-develop-a-convolutional-neural-network-from-scratch-for-mnist-handwritten-digit-classification/>
- <https://www.pyimagesearch.com/2020/08/24/ocr-handwriting-recognition-with-opencv-keras-and-tensorflow/>

*** The biggest task for sprint 2 will be creating a sequential model to recognize the data. This will likely spill over into sprint 3 because the understanding will be the most tricky part.

Sprint 3 (3/26 – 4/8)

Before sprint 3, we definitely need to come to a final consensus on whether we are making a web app or a mobile app – from our last meeting we discussed that we were leaning towards a mobile app on Android because it would be easier and more practical to take a picture from a phone or mobile device. For the second half of sprint 3, after the model is finalized, we should plan to serialize it and load it into the backend for our application.

- Finishing up model (see sprint 2 resources / plan)
- Serializing the model and loading it into a backend
 - TF docs on how to save and load keras models:
 - https://www.tensorflow.org/guide/keras/save_and_serialize
 - https://keras.io/guides/serialization_and_saving/
 - Deploying model with flask:
 - <https://towardsdatascience.com/deploying-keras-deep-learning-models-with-flask-5da4181436a2>
 - <https://curiously.com/posts/deploy-keras-deep-learning-project-to-production-with-flask/>
 - Deploying model to Android:
 - <https://stackoverflow.com/questions/45874245/keras-deep-learning-model-to-android>
 - <https://heartbeat.fritz.ai/deploying-pytorch-and-keras-models-to-android-with-tensorflow-mobile-a16a1fb83f2>
 - <https://towardsdatascience.com/how-to-convert-from-keras-to-tflite-with-zero-knowledge-of-tensorflow-5448a296ae67>

Sprint 4 (4/9 – 4/22)

- Make sure model is serialized and working properly
- Work on a front end
 - Create some sort of interface through which a person can take a picture
 - The picture should be passed somehow for the model to interpret it
 - Plan on returning some sort of page which either displays the written equation
- Can discuss working on logic for the equation to be solved.

Sprint 5 (4/23 – 5/6)

- Leftover – sprint 5 will be up in the air for now, what we do here depends on how much we get done in sprints 2 – 4. We will likely still be working on front end and back end interaction, putting the finalizing touches on project.
- Preparing for presentation