## **Release Plan**

Product Name: Neural Network GUI

Team Name: The Tensors

Release Number: ReleasePlan\_v01

Release Date: 4/5/2022

Revision Number: 2

Revision Date**:** 4/15/22

## **High level goals** (Ordered by priority)

* Be able to drag and drop modular blocks that represent neural network layers or tensor operations (to be abstracted into a single block)
* Be able to download auto-generated code that implements the designed network in common interfaces (such as pytorch and tensorflow)

## **User stories**

**Must have**

1. As a user of the website, I want a GUI for creating perceptron tensorflow networks, so that I can make them in an intuitive fashion.
2. As a user of the website, I want the GUI to allow for convolutional networks as well, so I can generate more types of networks.
3. As a user of the website, I want a GUI to also create pytorch networks, so that I can choose different implementations for the same network.
4. As a user of the website, I want to be able to download the code of the neural network that I designed.

**Should Have**

1. As a user of the website, I want to be able to access my neural networks across multiple machines.
2. As a user of the website, I want to generate the code that optimizes the network as well, as I want an intuitive interface for the entire machine learning pipeline.
3. As a user of the website, I want it to tell me basic information about the network, such as number of parameters and number of layers, so I can have a better understanding of the network at a glance.
4. As a user of the website, I want operator blocks for abstractions of common architecture patterns, so I can create and visualize networks based off of these patterns easier.

**Could Have**

1. As a user of the website, I want to be able to save the entire network and abstract it into a single block, so I can use this block for more complicated networks.
2. As a user of the website, I want to be able to save the graphical network, so I can leave the website and improve it later.
3. As a user of the website, I want some pre-made networks available, so that I can build off of common network templates.
4. As a user of the website, I want a basic tutorial to guide me on creating a simple network, so that my first time using this can go smoothly.
5. As an educator, I want the website to display some basic information about neural networks, so that this website can be a teaching tool.
6. As a user of the website, I want the generated networks to perform preprocessing on the input data (photos), so I can use the generated networks for a wider variety of image sets.
7. As a user of the website, I want the model I create to be run on a real dataset, so I can get a sense of the efficiency of my model
8. As a user of the website, I want to have multiple canvas’s open at the same time, in one tab.

**Tasks for Each User Story**

* 1. Create drag and drop functionality
  2. Implement code generation for tensorflow (dense)
  3. Connect backend and frontend
  4. Add ability to reset canvas
  5. Add ability to modify each block’s parameters

1. 1. Implement code generation for tensorflow (convolutional)
   2. Add backend/frontend integration for the convolutional module
2. 1. Compare the tensorflow library and the PyTorch library and find the similar parameters for the different layers
   2. Implement code generation for PyTorch (dense)
   3. Implement code generation for PyTorch (convolutional)
3. 1. Implement the congregation of the code generation for all the connected modules on the canvas, to create a complete neural network codebase
   2. Implement a way to change between PyTorch and tensorflow code generation
   3. Open downloaded code in a new tab
4. 1. Host the website through a hosting provider (insert provider)
   2. Add the ability to create accounts
   3. Add the ability to save neural networks on accounts
5. 1. Write optimization code for the tensorflow modules
   2. Write optimization code for the PyTorch modules
6. 1. Add a “at a glance section”
   2. To this section add input size, output size, # of layers
7. 1. Research common architectures
   2. Implement these architectures in code
   3. Add blocks for these architectures
8. 1. Add a custom module to the list of modules
   2. Add the ability to store a neural network as a module
   3. Add persistent remembrance tied to the account
9. 1. Add the ability to store the neural network in cookies of the browser
10. 1. Find common neural networks
    2. Implement these neural networks using our modules
    3. Save these neural network configurations
    4. Implement buttons that will generate these configurations on the canvas (not a custom module)
11. 1. Plan out the tutorial
    2. Add popups the first time a user visits the page on how to use the program
12. 1. Add data figures (input / output size)
    2. Add the efficiency of the network
    3. Add warnings about the network (if the input/output sizes are not correct for the module, etc)
13. 1. Figure out preprocessing algorithm
    2. Add preprocessing to the dataset depending on the size of the user’s input and output
14. 1. Implement sample input datasets
    2. Implement ability to run neural networks on these datasets
15. 1. Add multiple workspaces in the same tab
    2. Add multiple instances of the canvas running on the same tab
    3. Make sure that the multiple instances are not changing the state of other instances of the canvas

Doing A involves work that makes all of the rest possible, so it will probably be the hardest

Story points, fibonacci scale 1 2 3 5 8 13 20

| User Story | Story Points |
| --- | --- |
| A | 13 |
| B | 5 |
| C | 8 |
| D | 13 |
| E | 2 |
| F | 5 |
| G | 3 |
| H | 3 |
| I | 3 |
| J | 3 |
| K | 5 |
| L | 5 |
| M | 8 |
| N | 8 |

## **Sprint 1**

A

B

E

## **Sprint 2**

C

D

F

## **Sprint 3**

G

M

L

## **Sprint 4**

K

N

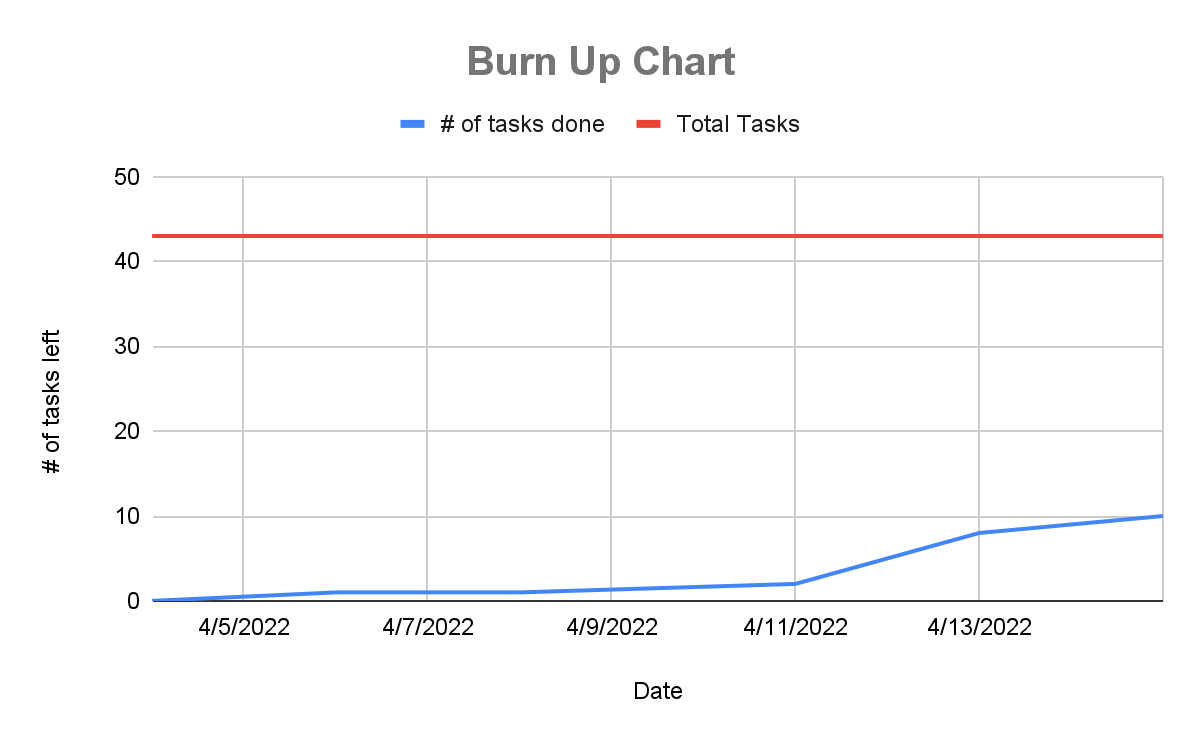
I

## **Minimum Viable Product**

## **Scrum Board**

Trello for a project management tool

Github Projects

****