

User Guide



CA400

GUI for learning about Neural Networks

Student Name: Liam McAweeney

Student Number: 1415152

Project Supervisor: Charlie Daily

Table of Content

Table of Content	2
Installation	2
User Guide	3
Login/Register	3
Home	4
Design	5
Train	8
Analysis	11
Classify	12
Logout	14

Installation

To deploy the project to a local machine for development there is just three prerequisites:

1. The latest version of Node is installed on your machine
2. You have Python 2.7.15+ installed on your machine
3. You have SQL installed on your machine

Once you have the two from above you are ready to download the source code. The project is available for download from my GitLab Repository:

<https://gitlab.computing.dcu.ie/mcaweel2/2019-ca400-mcaweel2.git>

You can download it directly from the url above or clone it from the command prompt with the following:

```
git clone https://gitlab.computing.dcu.ie/mcaweel2/2019-ca400-mcaweel2.git
```

The next step is to set up the database for storing test data. The easiest method for this is using workbench as it is simpler than creating a schema from the command prompt. Once you have created a schema, can file it the credentials for it in the setup.py file loaded at:

2019-ca400-mcaweel2/src/backend/setup.py

Below is a sample configuration for the setup file, you will also have to enter your email credentials for performing the verification of user's emails.

```
SQLALCHEMY_TRACK_MODIFICATIONS = False
SQLALCHEMY_DATABASE_URI = 'mysql://admin:testDBPassword!@localhost/Neural_network_users'
CORS_HEADERS = 'Content-Type'
SECRET_KEY = 'secret'
DEBUG = True
MAIL_SERVER = 'smtp.googlemail.com'
MAIL_PORT = 465
MAIL_USE_TLS = False
MAIL_USE_SSL = True
MAIL_USERNAME = 'email@gmail.com'
MAIL_PASSWORD = 'password'
```

Now it is time to install all related packages and node modules. I recommend the use of a virtual environment to remove any risk installation errors on your device. Venv is a perfect package for doing so. To install the packages on the backend you can enter the following command in the command prompt: `pip install -r requirements.txt` . To install the node modules on the frontend you can enter the following: `npm i`.

Once this is completed you are ready to host the application locally. Run: `./app.py`. This will start up the backend. To start up the frontend enter the following command: `npm run webpack-dev-server --mode development`. The web application is then available from the <http://localhost:3000/>

Congrats, you have success installed the application locally.

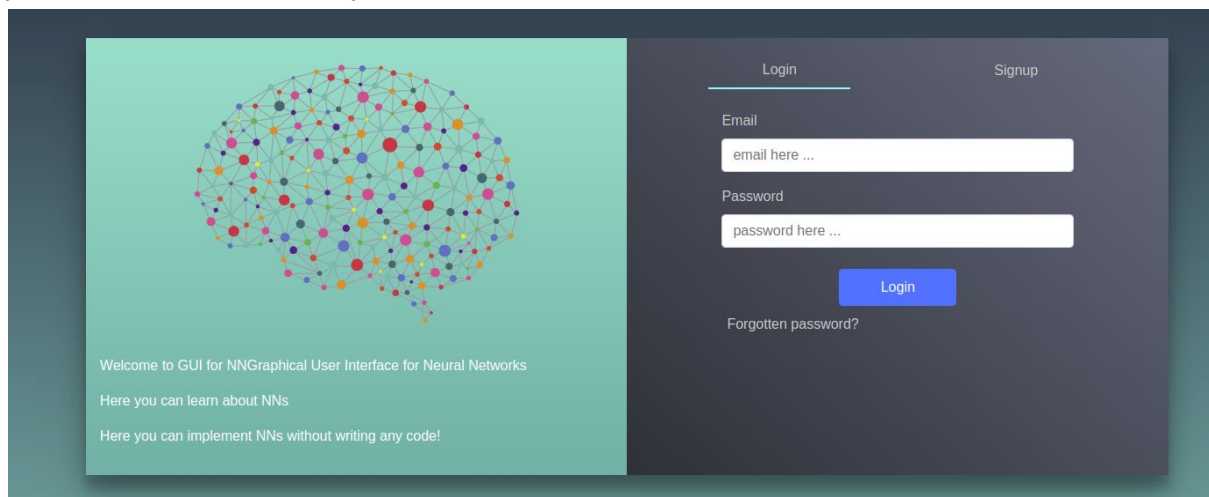
When editing code normally you must save it then restart the host to see the changes. React created a really useful library called react hot loader. It allows you to edit you code and save it, once it is saved it will automatically update the browser without having to reload the page or restart the host.

User Guide

Welcome to the user guide for this application. The application is very simple to use and easy and intuitive.

Login/Register

When you first reach the application you have to enter login or register as a user, either way you are prompted to enter you email address and password.

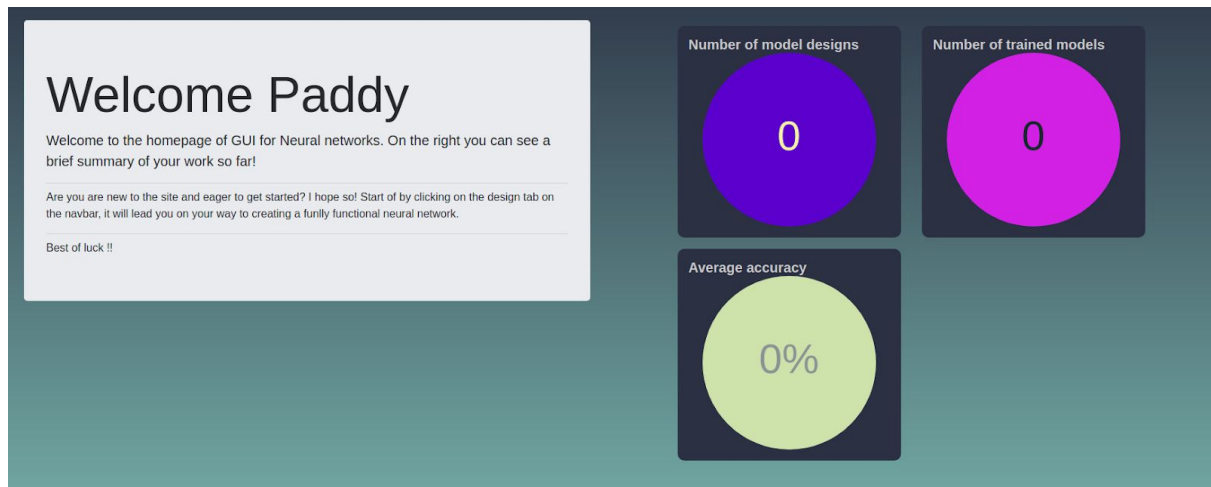


If you are registering as a new user you will be automatically sent an email with a link to follow to verify your email. After you have entered your password again for confirmation you are a verified user and must then enter your name. You name is only used to make the application more personalised for you.

Now you are logged in and are on the homepage of the application.

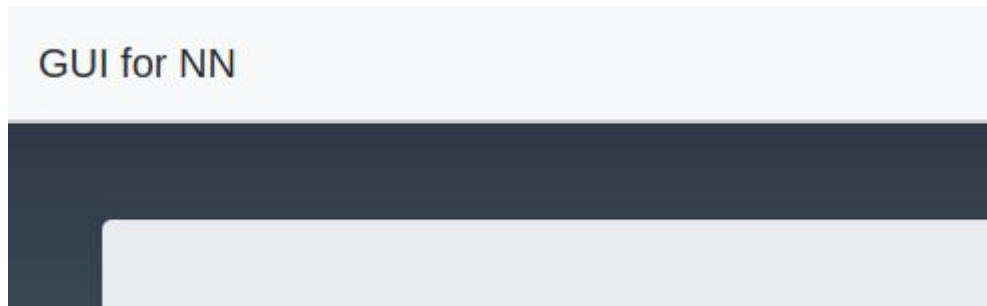
Home

The home contains a brief introduction on the left side of the screen. It also contains a summary of your progress on the application, ie. then number of models you have created and the average accuracy you achieved across your all you models. As you would assume for someone who is just after signing up these numbers are all zeros.

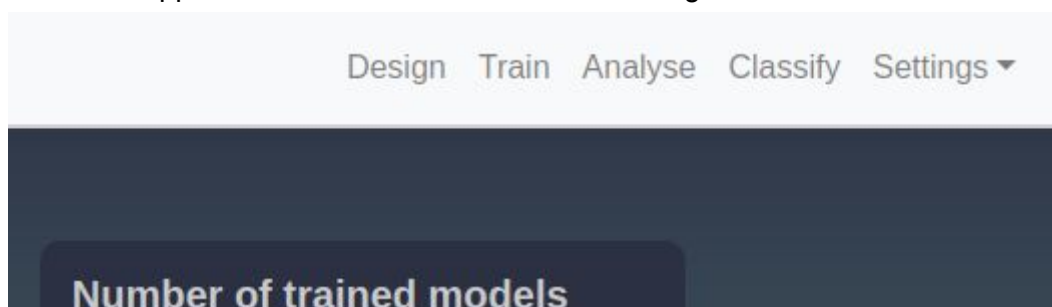


From here you have access to all the main components of the application. Along the top of the page is a navbar which links you to each component. The navbar is uniform across all of the application to help you familiar yourself with the application easily.

To access the homepage when you are exploring the application simple link on the name of the application located on the left side of the navbar. When you hover across this you will see that it is clickable as it will slightly change in colour.

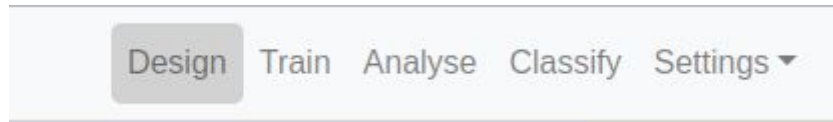


The same applies for each of the other tabs on the right side of the navbar.



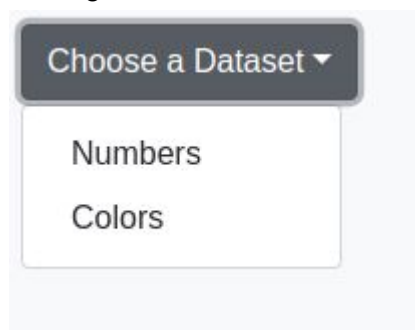
Design

You are now in the design section of the application, you will see that the design tab on the navbar is highlighted to ensure the user knows exactly where they are in the application. This is same no matter what section you are in.

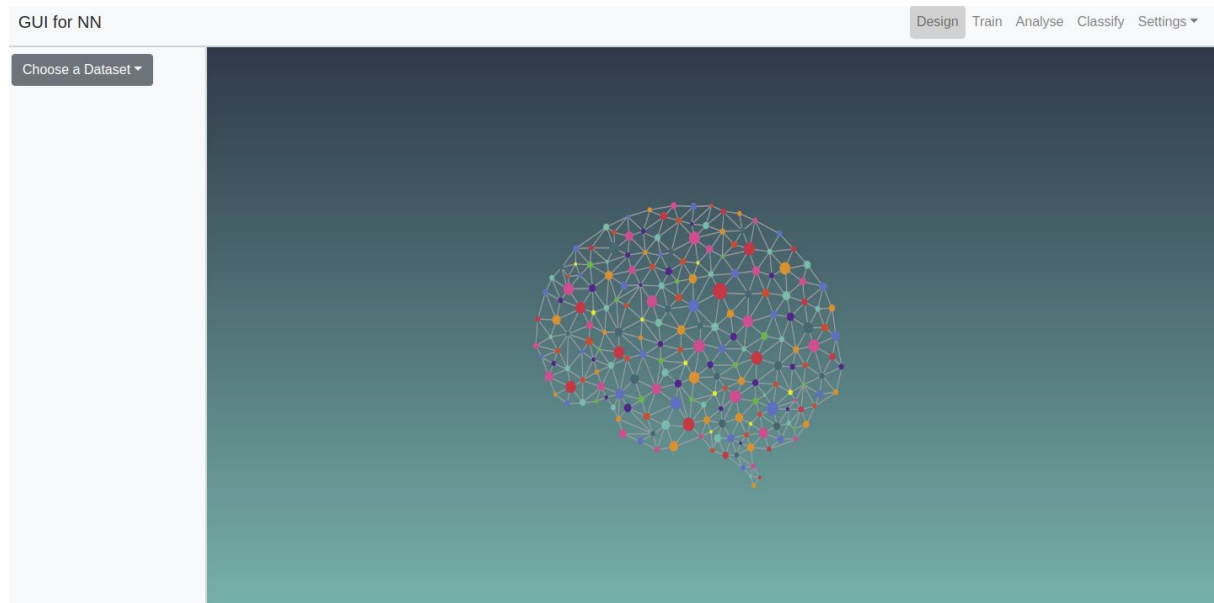


The design section is the first step in creating a neural network model. You will notice that there is a sidebar now on the screen. For each of steps involved in creating a model a sidebar will be present. This is to provide consistency to the user and in turn promote mastering the application.

The sidebar now contains a down down menu which contains the datasets available for training a model with.

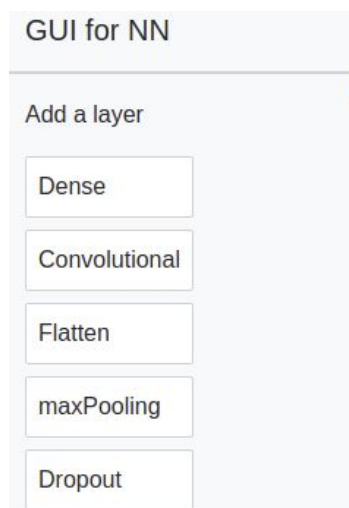


The next stage will depend on what dataset you select. The Numbers dataset is a dataset with 55,000 images with 0 and 9. The model architecture for this dataset could be a convolutional neural network. The color dataset is is a dataset with different color, 3 values representing RGB and a label for the color. This model design for this dataset can't be a convolutional so the next stage will have different options. While the user hasn't selected a dataset you can see the logo for the application floating in the main display. This is to reassure the user that the system is responsive.

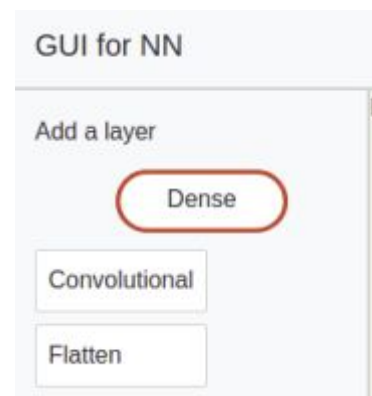


When the user selects the dataset for their model the sidebar is updated with the next step. The sidebar now contains all the different types of layers the user can put into their models. The user can select the type of layer they want to add in and drag it into the display. When you pick the layer up it a red border will be add around it to make it clearer.

Layers to add to the model



A layer while being dragged



After the layer is dropped into the model display, a modal will pop up for you to add some configuration for that layer.

Initial Number of Nodes

Dense

Number of Nodes

nodes here

Activation Function

eg: relu

Continue

For a dense layer you the number of nodes you want on that layer and the activation function you want.

Convolutional

Kernel size

eg: 3

Activation Function

eg: relu

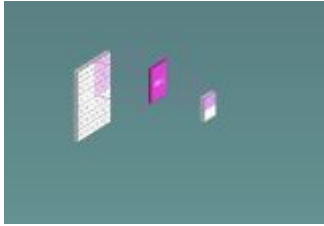
Number of filters

eg: 8

Continue

For a convolutional layer you have a different configuration.
For max pooling and dropout you also have different configurations. For a flattened layer there is no configuration as you are simple converting the model to the output from the previous layer to a 1D matrix.

Each layer is represented by a different image in the display, the example below is the image for a max pooling layer.



After the user has completed the design of their layer they must save it. On the sidebar there is a field where they fill in the name of the model and press save. An error message will appear if they have a model with that name already.

This model name is already used

Model Name

The model is then saved and it ready to be trained.

Train

When in the training section you will see the same sidebar again. This time it contains all the models you have designed so far.

Models ▾

After you select your model you will see the sidebar is once again populated with options. These options are for the training, ie. Number of epochs.

Number of epochs

Number of epochs

Size of batches

Size of batches

Number of Inputs: 1 - 55000

Number of Inputs

Validation Split: 0 - 100

Validation Split

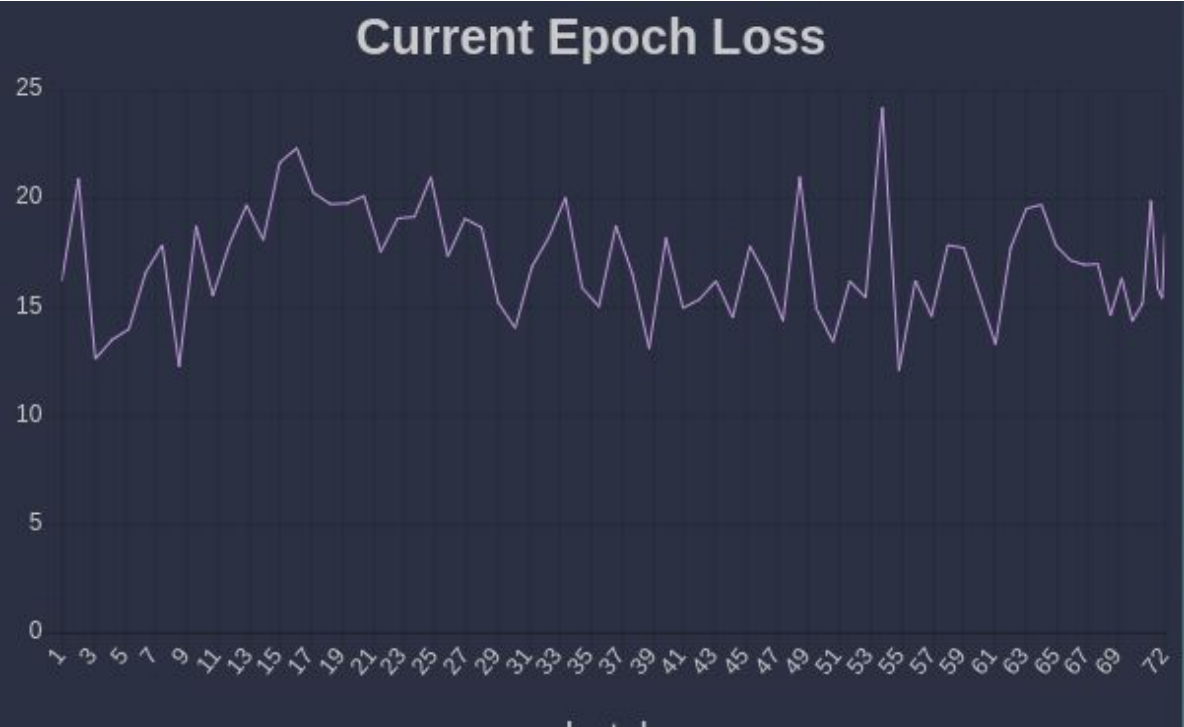
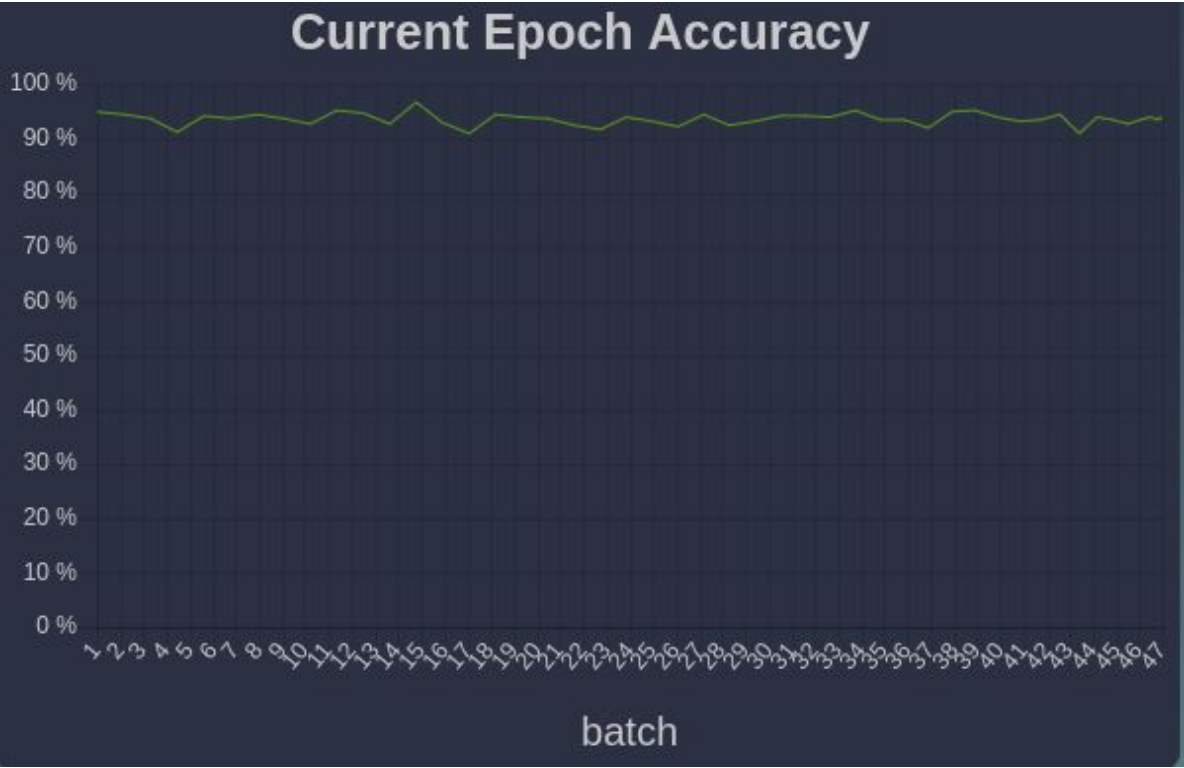
Train Model basic

Also there will be a short summary of your model's design in the display before you start the training. This is to ensure you have selected the model you wanted to.

After you have configured your model and are ready to train, hit the train button.

This will start the training of your model. You will see a chart pop up on the display which is rendering realtime information about your model's training.

There are 4 charts: current epoch's accuracy, current epoch's loss, overall accuracy and overall loss. Sample output as displayed below.

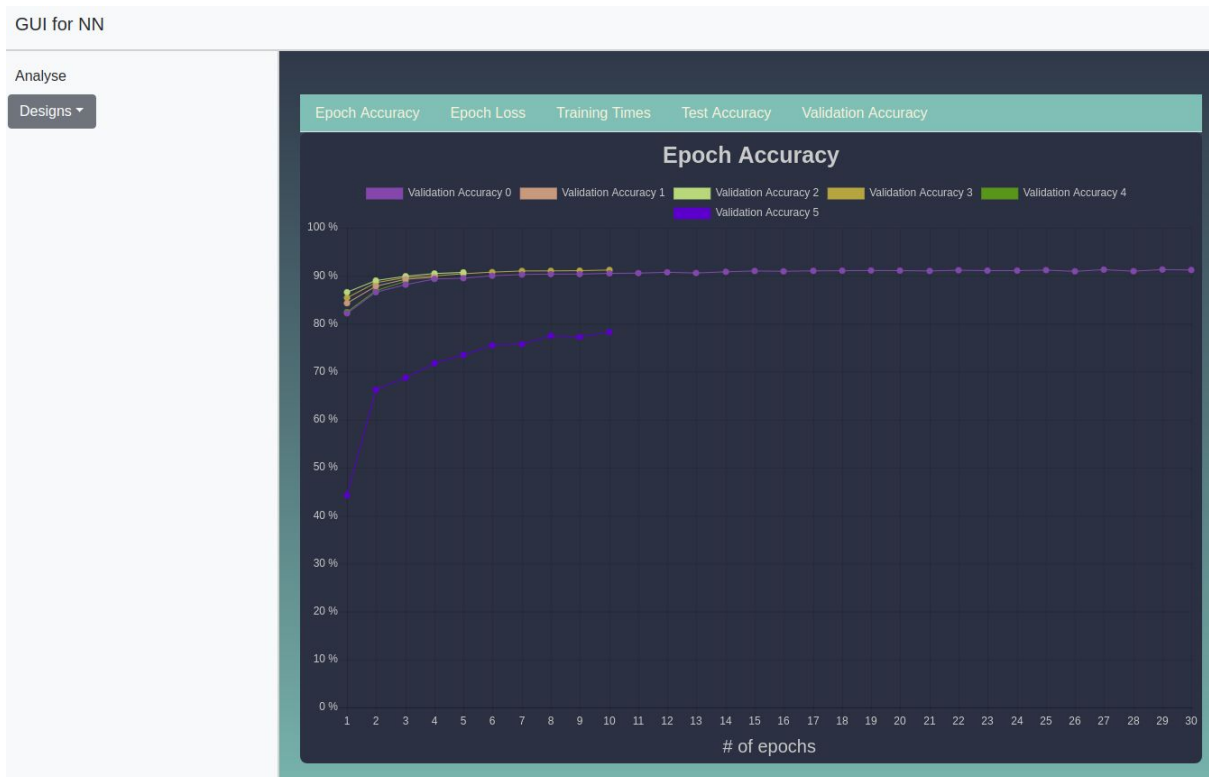




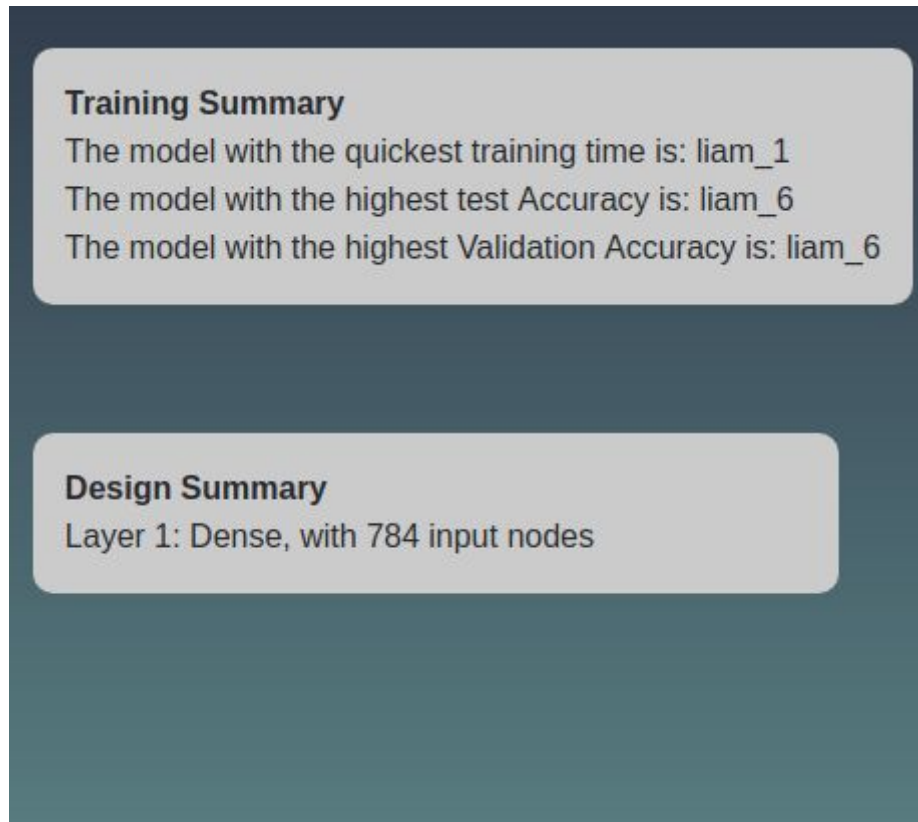
There is a tab created containing each of the epochs batch accuracy and loss for further analysis.

Analysis

In the analysis section you can view models you previously created and analyze their outputs. The same drop down option is on the sidebar, you select your model and the charts will be populated.



There are currently 5 different charts for analytics, each bringing an extra dimension to facilitate more learning. Again there is a short summary of the design of your model. There is also a brief description on how differently trained versions of the selected design are performing.



Classify

The last section in the application is the classification section. Here you can select the design from the sidebar, then choose which trained version of it you want to use for classifying.



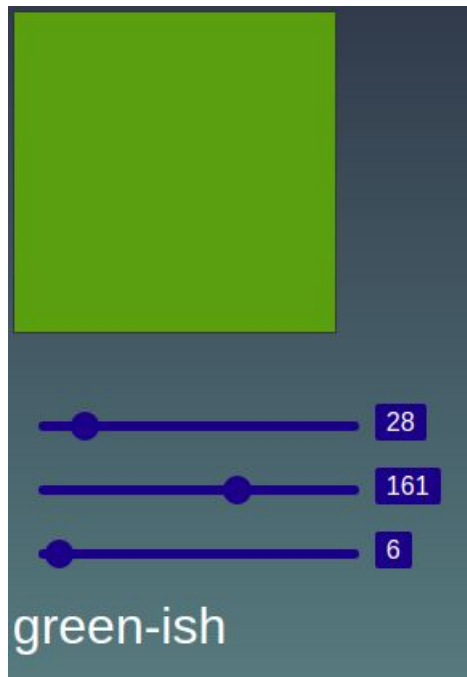
A summary of each model is output and you simply select the model you want to use.

Once selected you can in put draw a number into the canvas provided and press the button to predict.



Above is a sample output from model I trained.

If you have trained a model for the color dataset you can create a colour using the railings. The prediction for this model happens in real time thanks to TensorFlowTS



Congrats, you have now completed a full model and used it in a classification.

Logout

The last function is the logout function and it is locked in the settings dropdown.

