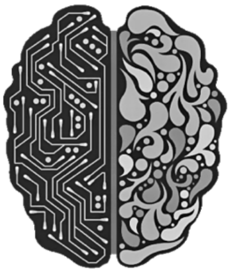
UNIVERSITY OF LIVERPOOL

Neural Network Visualization

Requirement Analysis



COMP 208: Group Software Project

Group 41

Group members: Geyang Xu, Haihang Hong, Jiacheng Shi, Jiayun Kang, Longxi He, Zetao Qian

Geyang Xu HaihangHong Jiacheng. Shi JiayunKang
Longxi He ZetaoQian

1. Project Description

1.1. Mission Statement

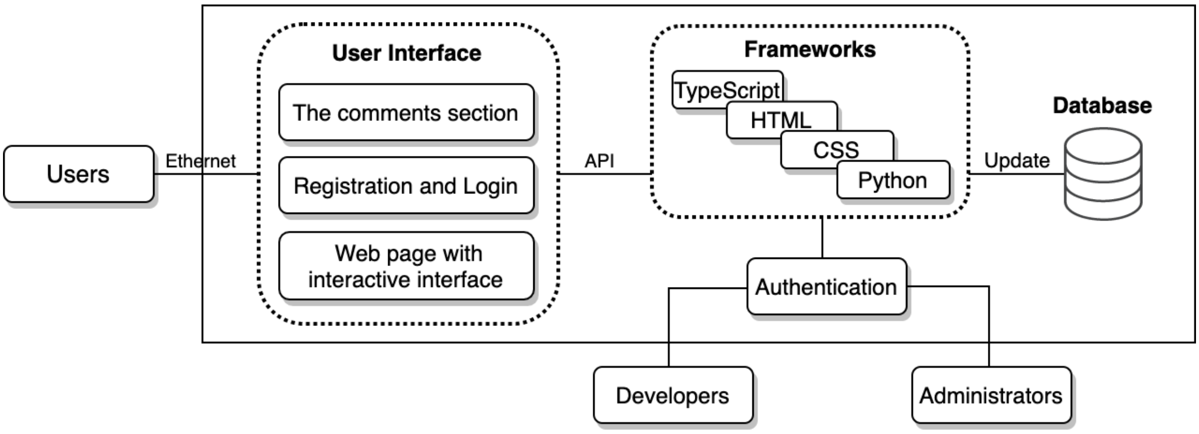
This neural network visualization website to help people who learn neural networks quickly understand and master the concept of neural networks and can also be used as a teaching aid tool for teachers in teaching.

1.2. Mission Objectives

This web project is designed to satisfy the people’s growing demand of learning and discussion neural networks on personal device in the Covid-19 pandemic setting. Below are the objectives of this system:

- To provide different styles of network architecture
- To represent structured information via abstract graphs and network diagrams
- To check the network structure, as well as changes of loss, accuracy and other indicators
- To visualize the structure of the network by dragging the appropriate parts
- To present the results of visualization in the neural network as well as the content of the network structure
- To provide a discussion area for learners to discuss

1.3. System Boundary Diagram



1.4. User Views & Requirements

1.4.1 User Views: [1]

Size of user group	All students who studied Computer Science or, professors who taught Computer Science
Age range	18+
Gender	Male/Female
Language	English
Education	Undergraduate/Postgraduate
Special skills	To have some basic Computer Science knowledge
Potential requirements	Design system to be easy to use, clear, and understandable, allowing users to navigate it easily

1.4.2 User requirements:

- To register and log in

Description	Before entering the website, the user must register an account, and then he/she can login to his/her account to go to the website to learn. Each user has his/her own account which includes his/her personal information.
Input	1. Users clicks the "Register" button to register. 2. Users enter his/her personal information, then clicks on the "Submit" button to register successfully. 3. Users fill in his/her account information, then click "Login".
Output	1. The default page is the login interface. User needs to enter the username (email) and password to log in. The "Login" and "Register" buttons are at the bottom. 2. After user clicks "Register", page turns to the "Create an account" page which requires the user to enter his/her name/email/password... 3. After user "Submit" his/her registration, the page is automatically redirected to the login page. 4. After user fill in his/her account information and click "Login", the page jumps to the main page.

- To run the deep learning model

Description	User can see the dynamic movement of deep learning model
Input	Users click the "play" button/ "pause" button/ "reset" button.
Output	1. After user clicks the "play" button, the image of deep learning model starts to run, and the "play" button turns into "pause" button; Show the test loss and training loss. 2. After user clicks the "pause" button, the dynamic image stops, and the "pause" button turns into "play" button. 3. After user clicks the "reset" button, all data and images are reset.

- To adjust the data/parameter/dataset of the deep learning model

Description	User can adjust the parameter/dataset of deep learning model
Input	1. User chooses dataset/learning rate/activation/regularization /regularization rate/problem type. 2. User clicks "+" and "-" to set the ratio of training to test data, noise, batch size, and the number of neurons and hidden layers. 3. User regenerates the data he/she has already set.
Output	1. The selected dataset is colored; the other unselected are gray. 2. According to the user's selection, display a certain number of neurons, hidden layers, and other data. 3. After user click regenerate, all the data in the image resets.

1.5. Transaction Requirements

- Data entry: Add the details of a new user
- Data update/deletion: Update/Delete the information of users
- Data queries: List the choices of dataset/learning rate/activation/regularization/regularization rate/problem type
- Constraint: User can only choose the dataset/activation provided.

1.6. Systems Specification

1.6.1. Server requirements

- Initial Database Size & Rate of Growth: The initial size is 100 tuples, and the capacity will be expanded according to the actual situation.
- Performance: Process a general incoming request in 50ms
- Security: the system ensures that personal information will not be leaked out in any way and service level agreement [2] up to two 9 when the system runs
- Back-up and Recovery: Website related codes and database are hosted by a third party to back up and control the version (GitHub private repository)

1.6.2. Client requirements

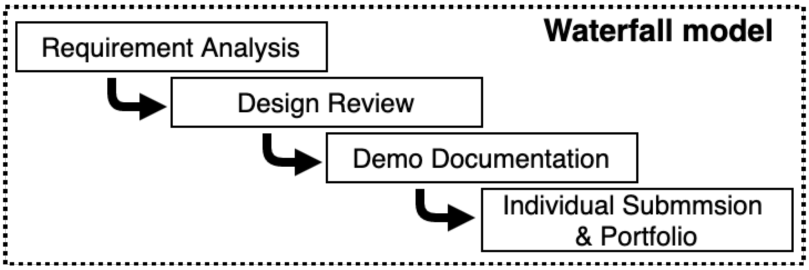
- The web will support any computers running Windows / Mac / Linux operating systems with web browser (Chrome, Safari & Firefox)
- Network: The client should have at least a bandwidth of 2M to browse this website normally

2. Conduct of the Project and Plan

2.1. Preparation (Background Research)

Through online surveys and group discussions, we decided to choose the knowledge and skills we have learned and the existing mature framework to complete this project. Therefore, we choose Django as the web framework, and the back-end language is Python. The front end chooses the current mainstream languages such as HTML+ CSS + TypeScript as the language for page structure design, page style design, and page behavior design respectively. Static types can facilitate teamwork and reduce communication costs. The database uses MariaDB, as an open-source branch of MySQL, with its ease of use and many related materials, which is very suitable as the database in the project. We have written a tiny neural network library for Neural Network Visualization website based on the TensorFlow [3].

2.2. Design stage



We designed a waterfall model to help team members understand the entire project process and follow.

Design documentation:

- Use Case Diagram
- UML Class Diagram
- Entity Relationship Diagram

2.3. Implementation stage

IaaS provider: DigitalOcean (London)

Hardware: 1 GB 1 CPU 25 GB SSD Disk

System: Ubuntu 18.04 (LTS) x64

Domain: team41.com

Frameworks: Django 3.1.7

Web Server: Caddy 2.3.0

Software:

IDE: Visual Studio Code & WebStorm
Version control: Git (Github.com)

Language:

Front-end: HTML + CSS + TypeScript
Back-end: Python 3.7

Database: MariaDB 10.5.8

2.4. Software Testing Methods

2.4.1. Black-Box testing [4]

- Equivalence Partitioning
- Boundary Value Analysis
- Causal-Effect Graphing

2.4.2. White-Box testing

- Statement Coverage
- Branch Coverage
- Path Coverage

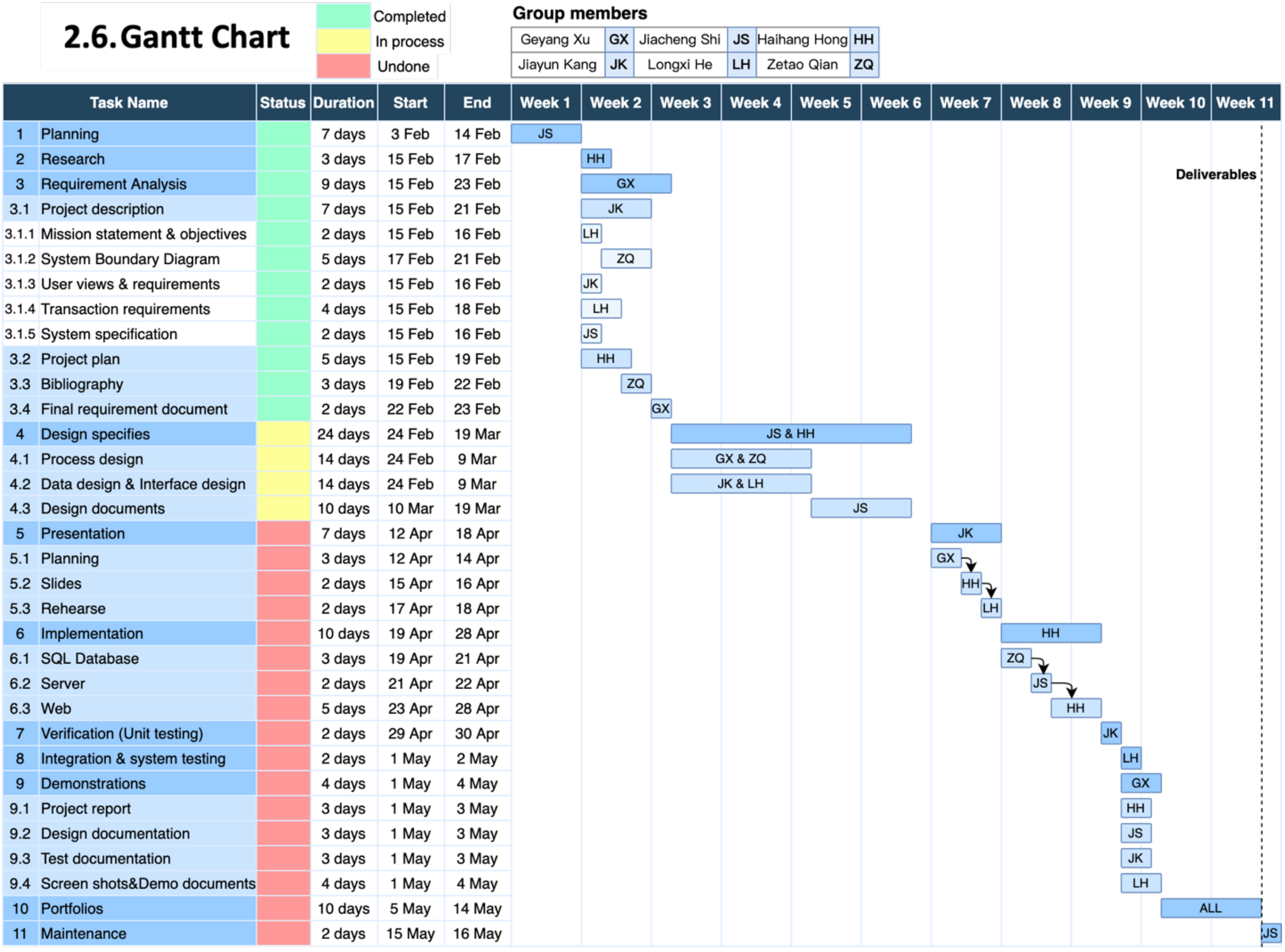
2.5. Risk Assessment

2.5.1. Project Challenge

- It is a new field for us, and it is difficult to assess the difficulty of the project.
- The team writes code together, and each module is merged with each module
- Fully test the program and make sure there are no errors

2.5.2. New Skills Required

- Front-end language programming for a website
- Remote connection server (SSH) and use of basic commands of Linux
- New skills will be available from the Internet and books



2.6. Gantt Chart

🟩	Completed
🟨	In process
🟥	Undone

Group members

Geyang Xu	GX	Jiacheng Shi	JS	Haihang Hong	HH
Jiayun Kang	JK	Longxi He	LH	Zetao Qian	ZQ

	Task Name	Status	Duration	Start	End	Week 1	Week 2	Week 3	Week 4	Week 5	Week 6	Week 7	Week 8	Week 9	Week 10	Week 11
1	Planning	🟩	7 days	3 Feb	14 Feb	JS										
2	Research	🟩	3 days	15 Feb	17 Feb		HH									
3	Requirement Analysis	🟩	9 days	15 Feb	23 Feb		GX									
3.1	Project description	🟩	7 days	15 Feb	21 Feb		JK									
3.1.1	Mission statement & objectives	🟩	2 days	15 Feb	16 Feb		LH									
3.1.2	System Boundary Diagram	🟩	5 days	17 Feb	21 Feb		ZQ									
3.1.3	User views & requirements	🟩	2 days	15 Feb	16 Feb		JK									
3.1.4	Transaction requirements	🟩	4 days	15 Feb	18 Feb		LH									
3.1.5	System specification	🟩	2 days	15 Feb	16 Feb		JS									
3.2	Project plan	🟩	5 days	15 Feb	19 Feb		HH									
3.3	Bibliography	🟩	3 days	19 Feb	22 Feb		ZQ									
3.4	Final requirement document	🟩	2 days	22 Feb	23 Feb			GX								
4	Design specifies	🟨	24 days	24 Feb	19 Mar			JS & HH								
4.1	Process design	🟨	14 days	24 Feb	9 Mar			GX & ZQ								
4.2	Data design & Interface design	🟨	14 days	24 Feb	9 Mar			JK & LH								
4.3	Design documents	🟨	10 days	10 Mar	19 Mar					JS						
5	Presentation	🟥	7 days	12 Apr	18 Apr							JK				
5.1	Planning	🟥	3 days	12 Apr	14 Apr							GX				
5.2	Slides	🟥	2 days	15 Apr	16 Apr							HH				
5.3	Rehearse	🟥	2 days	17 Apr	18 Apr							LH				
6	Implementation	🟥	10 days	19 Apr	28 Apr								HH			
6.1	SQL Database	🟥	3 days	19 Apr	21 Apr								ZQ			
6.2	Server	🟥	2 days	21 Apr	22 Apr								JS			
6.3	Web	🟥	5 days	23 Apr	28 Apr								HH			
7	Verification (Unit testing)	🟥	2 days	29 Apr	30 Apr									JK		
8	Integration & system testing	🟥	2 days	1 May	2 May									LH		
9	Demonstrations	🟥	4 days	1 May	4 May									GX		
9.1	Project report	🟥	3 days	1 May	3 May									HH		
9.2	Design documentation	🟥	3 days	1 May	3 May									JS		
9.3	Test documentation	🟥	3 days	1 May	3 May									JK		
9.4	Screen shots&Demo documents	🟥	4 days	1 May	4 May									LH		
10	Portfolios	🟥	10 days	5 May	14 May										ALL	
11	Maintenance	🟥	2 days	15 May	16 May											JS

3. Bibliography

[1] Raschka, S., "Python machine learning," Packt publishing ltd,2015. [Online]. Available: https://books.google.co.uk/books?id=GOVOCwAAQBAJ&hl=zh-CN&source=gbs_navlinks_s

[2] "The Global IP Network for NTT Delivers High-Quality IP Servers," Gin.ntt.net, 2021 [Online]. Available: https://www.gin.ntt.net

[3] "An end-to-end open source machine learning platform," Tensorflow.org, 2021 [Online]. Available: https://www.tensorflow.org/learn

[4] STF, "Software Testing Methods," Sep 16, 2020. [Online]. Available: https://softwaretestingfundamentals.com/software-testing-methods