Version Control System for Deep Learning

Professor: Yug Yung Lee

Project Pre-proposal:

Team Details:

Team ID: 6

Name: Dinesh Kumar, Kusam

Class ID: 14

Name: Pradeepika, Kolluru

Class ID: 12

Name: Sindhusha, Tiyyagura

Class ID: 24

Name: Sravan Kumar, Pagadala

Class ID: 21

Project Goal and Objectives:

Motivation:

The Motivation of our project is to create simple and flexible git-like interface and architecture for Deep Learning projects.

Significance/ Uniqueness:

Existing Version control system(Git) is used to track a single file while our proposed system is also used to handle tracking set of files or folders required for deep learning projects.

Objectives:

- The main objective of our proposed system is to bring collaboration, reproducibility and agility into existing deep learning workflow.
- To maintain track of changes to an individual experiment or set of files so that one can review specific versions.

Scope of project:

Scope of our project is to reduce redundancy in version control and help users to keep an artifact clean, well-organised and allows flexible rolling back to previous versions.

System Features:

- Automatic grouping of set of artifacts(modeling artifacts, source code, data sets) into a single version.
- Command line interface compatibility
- Creating a new environment for each run of experiment.

Related Work:

1. Github:
   Github is a DVCS(Distributed version control system) used for tracking changes to the files and allows multiple developers to work simultaneously.
2. ModelHub: Deep learning Life Cycle Management
   In order to deal with rich set of artifacts, Modelhub adopts a modeling version control system, a domain specific language for seeking through model space and hosted service.
3. Towards Unified Data and Life Cycle Management for deep learning.
   This paper mainly concentrates on implementation of data and life cycle management system for deep learning. Firstly, proposes a high-level domain specific language to speed up the modeling process. Thereafter, a novel model versioning system and parameter archival storage system(PAS) are developed which reduces storage footprint.

Bibliography:

[1] Hui Miao, Ang Li, Larry S. Davis, Amol Deshpande, “ModelHub: Deep Learning Lifecycle Management” 2017.

https://par.nsf.gov/servlets/purl/10041785

[2] Hui Miao, Ang Li, Larry S. Davis, Amol Deshpande, “Towards Unified Data and Lifecycle Management for Deep Learning” 2016

https://arxiv.org/pdf/1611.06224.pdf

Project Plan:

Prioritized Features/Technologies:

1. Using python language for developing git-like VCS server.
2. Using SSH key mechanism to authenticate users.
3. Using DAG(Directed Acyclic Graph) data structure for handling objects/commits of files.

Project Increment 1:

User Story-1: Getting hands on with GIT source code.

- Download git source code
- Compile and run the code
- Change one of the module to get the feel of working with GIT

Estimated start and end time: 11th Feb 2019 to 20th Feb 2019

Assigned to: Task is to be done by all team members individually.

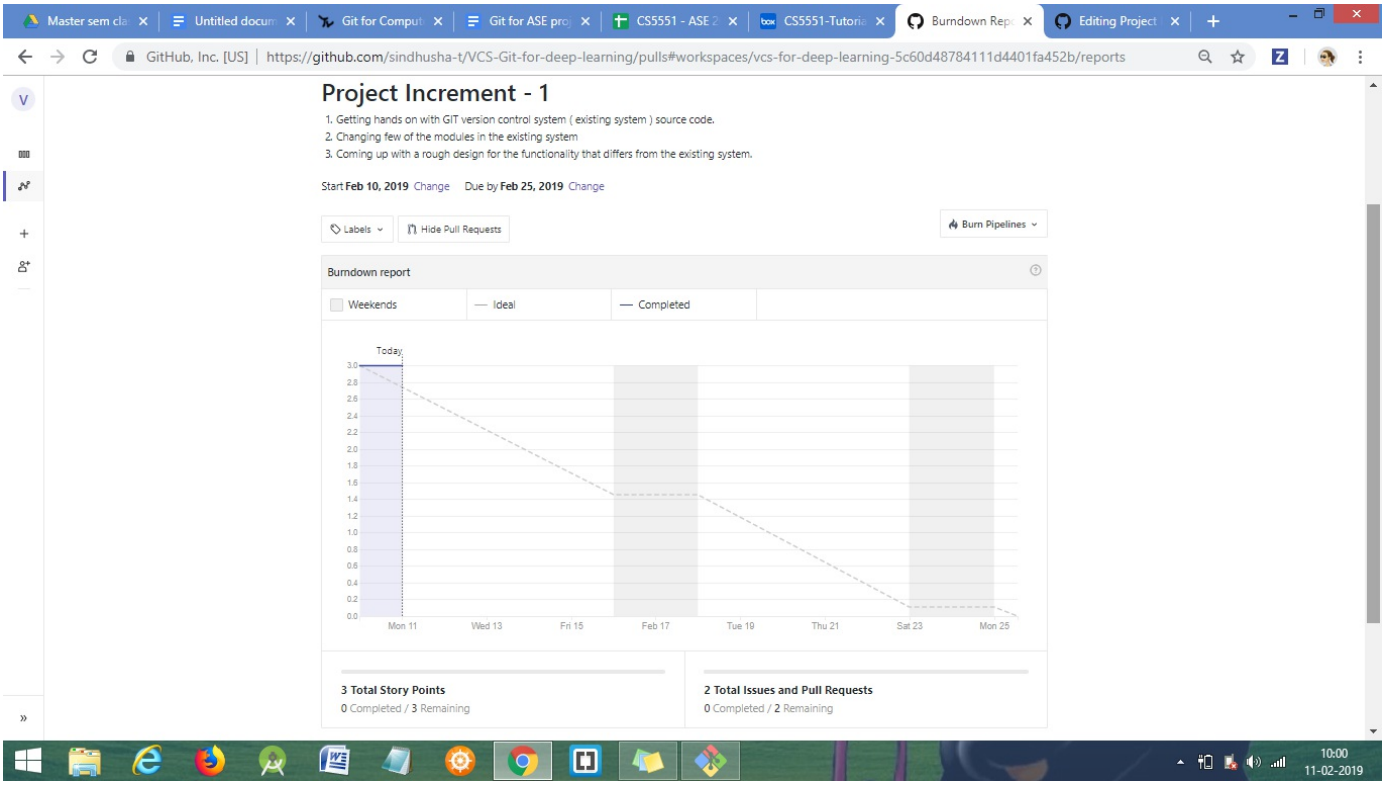
User Story-2: Understanding the design and coming with the rough design for the project

- The main difference between GIT version control system and the proposed system is in the functionality of tracking the files.
- In proposed system, modeling artifacts(models, data sets, source code) need to tracked as a single version, where as in GIT each files is tracked separately.
- Coming up with a rough design for the new proposed project.

Estimated start and end time: 20th Feb 2019 to 25th Feb 2019

Assigned to: Task is to be done by all team members individually and discuss to group all the ideas.

Burndown-chart:



Project Increment 2:

User Story 1: Creating Dummy DB server to store credentials and large data files

- Creating just a working environment of DB for the current project to work.
- As Storage of large data files will be taken care by another project.

Estimated start and end time: 25th Feb 2019 to 5th Mar 2019
Assigned to: Task is to be done by all team members individually.

User Story 2: User registration and authentication/login requests to be handled.

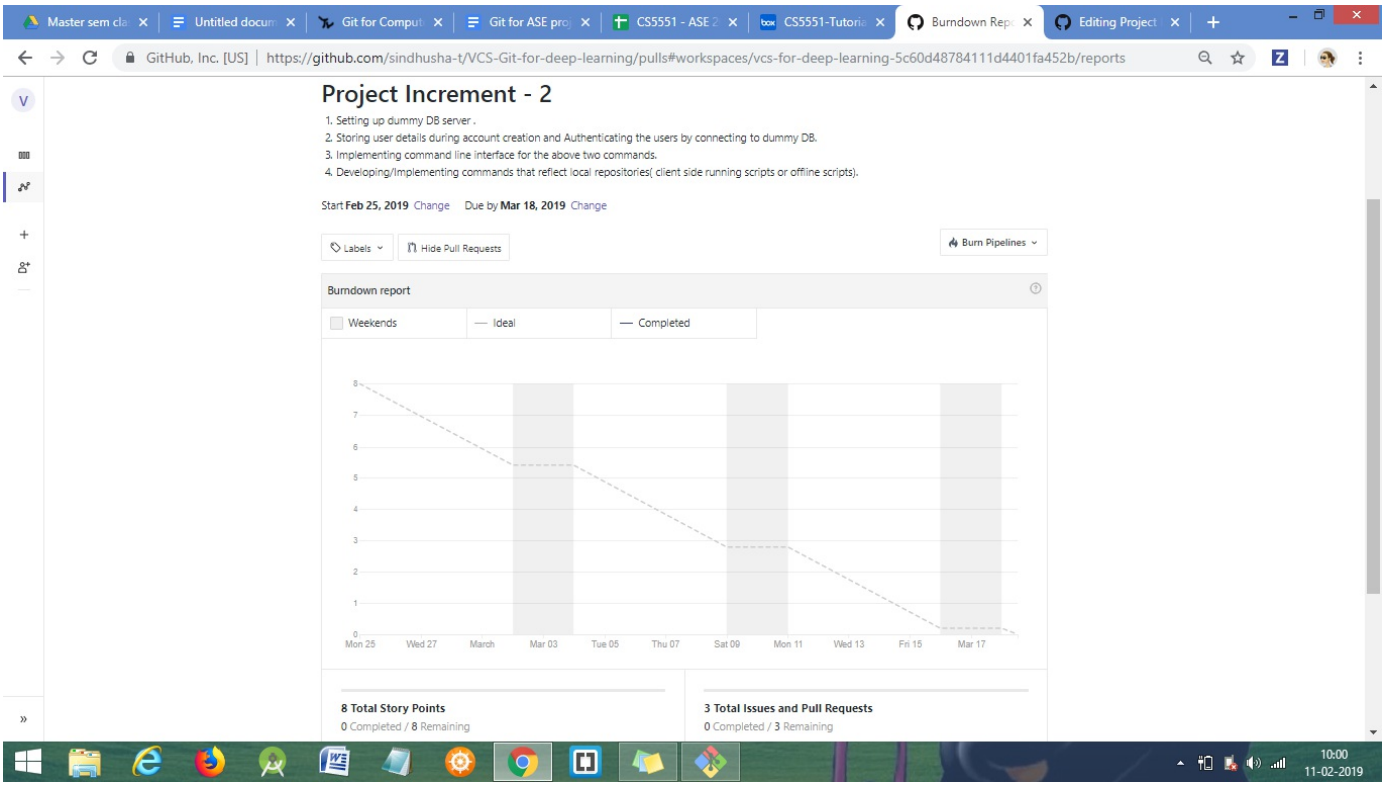
- Developing scripts to register a user and to authenticate the user.
- Should first connect to DB cache details, if there is a miss response then should connect to the proxy DB for user credentials.
- Enabling the scripts to work in the command line interface.

** Design Flow:**

1. Client either requests for authentication or login to the server.
2. Server connects to back-end database either to store or retrieve user details.
3. Checks validation and sends response to the client.

Estimated start and end time: 5th Mar 2019 to 18th Mar 2019
Assigned to: Sindhusha Tiyyagura

Burndown chart:



Project Increment 3:

User Story 1: Developing commands that reflect changes in the local repository(client) without contacting server.

- Commands that reflect changes only at the client side (AKA client side scripts/commands)

  gitdl init

  gitdl add

  gitdl remove

  gitdl commit

  gitdl tag

  gitdl branch

  gitdl checkout
- All these commands need to be changed to track artifacts as a single version. The functionality to be followed is similar to the

DAG(Directed Acyclic Graph) as of now. (which includes objects and references to the blob files)

Estimated start and end time: 18th Mar 2019 to 31st Mar 2019
Assigned to: Sravan Kumar Pagadala works on writing python scripts for init, add commands.
Pradeepika kolluru works on writing python scripts for remove, commit commands.
Dinesh Kumar Kusam works on writing python scripts for tag, branch commands.
Sindhusha Tiyyyagura works on writing python scripts for checkout command.

User Story 2: Developing commands that client connects to the server for the response.

- Commands that are made changes in the local repository are reflected at the server side.
  gitdl clone
  gitdl push
  gitdl pop
- Need to think of where to store the large files at the server side.

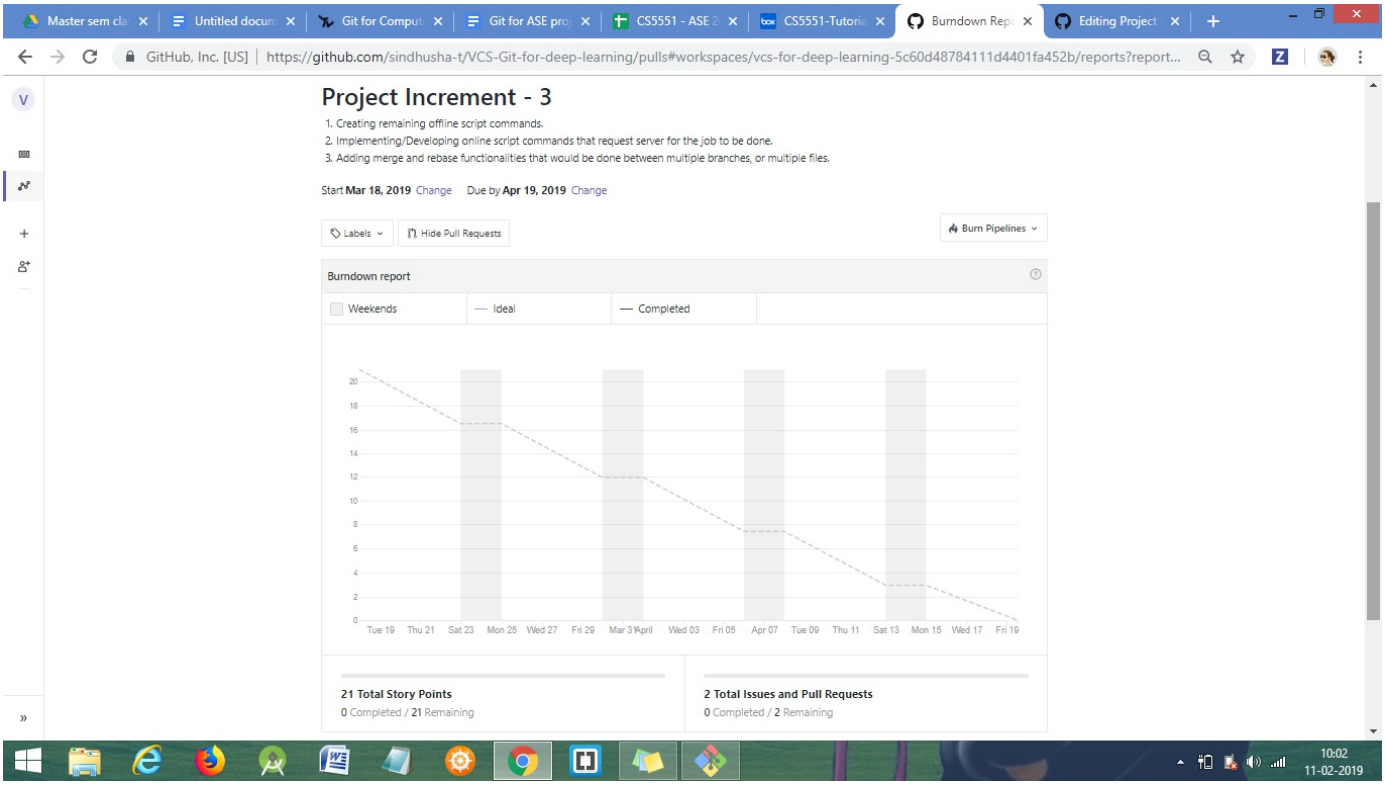
Estimated start and end time: 31st Mar 2019 to 19th Apr 2019
Assigned to: Sindhusha Tiyyyagura works on writing python scripts for clone command.
Pradeepika kolluru works on writing python scripts for push command.
Dinesh Kumar Kusam works on writing python scripts for pop command.

User Story 3: Giving the list of repositories in the user workspace.

Estimated start and end time: 31st Mar 2019 to 19th Apr 2019

Assigned to: Sravan kumar pagadala works on giving complete details of the user during the login stage.

Burndown-chart:



Project Increment 4:

User Story 1: Handling merging, rebasing, and diffing of set of files/ artifacts.

1. Adding the flexibility of adding and managing branches for the client.
2. Using tools for merging and diffing of file contents.

Estimated start and end time: 19th Apr 2019 to 25th Apr 2019

Assigned to: Will be done by Sindhusha and Pradeepika

User Story 2: Testing different functionalities in the project

Estimated start and end time: 19th Apr 2019 to 25th Apr 2019

Assigned to: Will be done by all team members for complete source code.

Burndown-chart: