Table of Contents

Freshers Training

Tools - 1

Git

1. Create a private Github repo. Any code, summary of what you read, etc. should be maintained in this repo throughout this exercise.
2. Demonstrate your understanding of the below git commands and their options. Each command had many sub-options. You can try the commonly used ones for each of the commands.
   a. git clone
   b. git add
   c. git diff
   d. git commit
   e. git status
   f. git branch
   g. git checkout
   h. git stash
   i. git pull
   j. git fetch
   k. git push
   l. git merge
   m. git rebase
   n. git cherry-pick
   o. git log
   p. git reset
   q. git tag
3. After doing the above, verify if you know about these git and its related concepts. Demonstrate with examples:
   a. Repository
   b. Branch
   c. What is a Remote in git? Create a few remotes for your repo and see the difference
   d. Difference between Local and Remote branches
   e. What happens when local and remote are not in sync?

f. What happens when you delete a local branch? What happens when you delete a remote branch?
g. What do these mean? Merge Requests and Pull Requests
h. What is a fast-forward merge?
i. What is a squashed merge?
j. What is a HEAD?
k. What happens when you try to merge branch *source* into branch *target*, but *target* is ahead?
l. What is the difference between Git and Gitlab (or Github)?

Linux Command Line

Show your understanding of the below concepts/commands:

1. Use of .bashrc file
    a. Create one on your own
    b. After changing the file, what should you do to make the changes work?
2. Environment variables
    a. What are the different ways to set env variables and how to set them?
    b. What is a *.env* file and how to use it to set env variables?
3. Demonstrate the usage of the below commands
    a. ssh - using password and ssh key separately
    b. scp - using password and ssh key separately
    c. rsync
    d. find
    e. grep
    f. wc
    g. head / tail / less / more
    h. which
    i. Commands to
        i. find your linux distribution
        ii. find linux kernel version
        iii. find number of Cores/Socket
        iv. check RAM size
        v. monitor currently running processes and their RAM/Core usage
        vi. sort running processes by their RAM/Core usage
        vii. check if your laptop has any GPU

        viii. find the number of disks in your machine and their usage
        ix. find all the USB devices connected in your machine
    j. Difference between sudo and super user
4. How to mount a disk to your machine?
    a. What is an mount point?
    b. What is an fstab file?
    c. How to unmount?
    d. Check the partitions in a disk
    e. What does the command `duf` do ?
5. apt - Why is it required and how do you use it?
6. Tmux
    a. Install tmux
    b. What is the difference between a terminal and tmux
    c. Learn different options in tmux and demonstrate them
    d. Write a .tmux.conf file according to your needs and explain the options

Tools - 2

Virtualenv

1. What is a virtualenv?
2. What is the difference between installing something inside a virtualenv and otherwise?

Pip

1. What is a package manager / installer
2. Software repository
    a. What are the available software repositories for Java, JavaScript, Python .. etc
    b. What is Pypi
3. Create a Pypi compatible package for some toy Python code of yours
4. Demonstrate different ways of installing your Python package
    a. using requirements.txt
        i. Write your own requirements.txt
        ii. Create requirements.txt using `pip freeze`
    b. using setup.py
5. What is the difference between the above two methods? Which one is preferred over the other in what situations?
6. Read about "site-packages". Find the difference between the contents of your package in the site-packages directory and your development directory
7. Figure out the difference between a whl file format and other file types for Python package
8. What is editable mode in a package installation?
9. What is a dependency conflict? How to handle when there is one?

Conda

1. Difference between Pip and Conda
2. Install Conda/Miniconda
3. Create an environment
4. Activate the environment
5. How to set environment variables when activating an environment?
    a. What happens to those variables when the environment is deactivated?
6. Install any package inside the environment
7. List down the packages installed inside the environment

8. Export the current environment to a file
9. Remove the package from the environment
10. Deactivate the environment
11. Delete the environment
12. Update Conda Version
13. Uninstall Conda

Poetry

1. Why poetry?
2. Installation
    a. Installing a specific version
3. Update poetry version
4. Add dependency
    a. Add a package from github repo
        i. From a specific branch or tag or commit
    b. Add a package from a local path
    c. Other possible ways to add a dependency
5. Poetry lock file
    a. What is a poetry.lock file
    b. Create a poetry.lock and install the dependencies (inside a conda env)
    c. Create a poetry.lock without installing any package
6. Update dependencies
7. Show poetry config
8. Modify poetry config values
9. Uninstall poetry

Black and Isort

1. What is a formatting tool
2. Install black and isort
3. Write a sample program and get it formatted using black and isort
4. How do you check
    a. If the code is already formatted or not
    b. If all the imports are properly ordered
    c. if there are any unwanted imports in the program

Mypy

1. Static vs Dynamic type checking
2. What is a type checker
3. Why we need one for Python
4. Write an example programs with static typing using mypy
5. How to skip/ignore an error?
6. How to skip file or code block from being checked by Mypy?
7. mypy.ini
    a. What are the contents of this file
    b. Create mypy.ini and add it to your repo

Pylint

1. What is Linting / Static Code analysis
2. Write an example program and run pylint over that file
3. How to skip/ignore an error that Pylint raises?
4. How to skip a file from being checked by Pylint?
5. pylintrc file
    a. What are the contents of this file
    b. Create pylintrc file and add to your repo

Pytest

1. What is a unit test? How is it different from a behaviour test?
2. Write few unit tests and run pytest
3. Get the report of those runs
    a. Save the report
4. Write sample pytest fixtures
    a. Reuse them inside uni tests
    b. Reuse them inside other fixtures
5. Use temporary directories and files
6. What is conftest.py
    a. Create a sample file
7. Mock objects
8. Catch raised Exceptions

DVC

1. Why do we need DVC ?
2. In your local file system, demonstrate your understanding of DVC by pushing/pulling the data.
3. How do you remove the data from local DVC cache alone ?
4. Suggest a workflow on how to maintain data in a ML experiment.

Python - 0

You'll go through a few lectures in CS50's Introduction to Programming with Python course.

Mandatory Readings

1. Week 0 - Functions
2. Week 1 - Conditionals
3. Week 2 - Loops
4. Week 3 - Exceptions

Exercises

Complete all the problem sets given as a part of each of the above lectures

Python - 1

Mandatory Readings

1. Week 4 - Libraries
2. Week 5 - Unit Tests
3. Week 6 - File IO
4. Week 7 - Regular Expressions
5. Week 8 - Object Oriented Programming
6. Week 9 - Et Cetera

Exercises

Complete all the problem sets given as a part of each of the above lectures

Python - 2

Mandatory Readings

- Fluent Python, Chapter - 1, "The Python Data Model"
- Fluent Python, Chapter - 2, "An Array of Sequences"
- Fluent Python, Chapter - 3, "Dictionaries and Sets"

Exercises

1. Demonstrate your understanding of a few "dunder methods" in Python
2. What is the difference between using "__repr__" and "__str__"? Explain with a code snippet
3. Implement and show the use of
   a. __getitem__, __setitem__, __delitem__, __contains__
   b. __len__
   c. __call__
   d. __enter__ , __exit__
4. What are the ways to initialize a list? Show them by implementing short code snippets
5. Show the difference between tuple and list
6. Show how to sort a list of tuples by ordering based on the second element of the tuple. Feel free to use built-in functions. ex: input: [(1, 2), (3, 6), (5, 1)] => output: [(5, 1), (1, 2), (3, 6)]
7. What are the new "dunder" methods you found for the new Mapping related types such as dict and set?
8. Show usage of defaultdict, missingitem and setdefault
9. Show your understanding of Counter and OrderedDict mentioned in Chapter 3
10. With example code snippets, show your understanding of the primary differences between set and list in Python

Python - 3

Mandatory Readings

- Fluent Python, Chapter - 14 "Iterables, Iterators and Generators"

Exercises

1. What is the return type of "range()" function?
2. Show your understanding of iterables vs iterators with code examples
3. What is the difference between yield and return?
4. When is yield useful?
5. Show the usage of __iter__ and __next__ methods
6. Implement the below
    a. Create a text file with 10 million lines
    b. Write a function that will return the list of all lines
    c. Write a function that yields the lines from the file
    d. Inspect the time taken and memory consumed for each of the above cases

Python - 4

Mandatory Readings

- Fluent Python, Chapter - 9, "A Pythonic Object"

Exercises

1. Demonstrate the usage of staticmethod and classmethod with explanations of when are they useful
2. How is access control on attributes of a class (public, private, etc.) different in Python vs other programming languages?
3. What is the difference between a class attribute and an instance attribute?
4. Learn about dataclasses and namedtuple
5. Show your understanding of a dataclass, namedtuple with code examples. How are they different from normal classes?
6. Learn about inheritance in Python and show with examples your understanding of the below concepts
    a. Basic inheritance
    b. Abstract class (see collections.abc module in Python std lib)
    c. Protocol (see typing.Protocol module in Python std lib)

Python - 5

Exercises

1. Learn and show your understanding of map, reduce, filter
2. Python functions are said to be "first class objects". What is the meaning of this? Show various cases where this is true
3. Write a lambda function and how is it different from a regular function?
4. Read about "higher order functions" and show a simple implementation to explain their usage
5. Write a simple Python decorator that would print the time taken by any function
6. How does Generics work in Python? Learn and show your understanding by implementing a snippet that uses Generics
7. Implement an example that uses overloading in Python. How is overloading different in Python from other languages and why?
8. What is the difference between a Python script, package and a module?

Python - 6

Readings

- HTTP Basics - MDN Web
- CS50 2017 - Lecture 6 - HTTP (YouTube)
- FastAPI Course for Beginners (YouTube)
- FastAPI Docs
- Working with JSON data in Python
- CS50 2020 - Lecture 7 - SQL (YouTube)
- CS50 - SQL, Models and Migrations (YouTube)
- CS50 - JavaScript (YouTube)

Exercises

Complete the below projects from CS50 Web course. Note that the starter code provided by CS50 is in Django (another Python Web framework). Your task is to understand the specifications given in the project description and implement everything in FastAPI instead of Django.

1. Project 1 - Wiki
2. Project 2 - Commerce
3. Project 3 - Mail

ML - 2

Backpropagation and Gradients

Readings:

- Calculus on Computational Graphs: Backpropagation
- CS231n Winter 2016: Lecture 4: Backpropagation, Neural Networks 1 (YouTube)
- The Fundamentals of Autograd (YouTube)
- PyTorch Autograd Explained - In-depth Tutorial (YouTube)
- Backpropagation with Numpy and PyTorch
- The spelled-out intro to neural networks and backpropagation: building micrograd - Andrej Karpathy (YouTube)

Exercises:

1. Write a few functions (linear, quadratic, etc.) and compute their partial derivatives by hand. You can assume the input to be scalar values

2. Write Python code to compute the gradients for the same functions with Numpy and verify if the results match with the ones you computed manually

3. Write Python code to compute the gradients for the same functions with PyTorch tensors without using PyTorch's autograd. Verify if the results match with the above ones

4. Use PyTorch autograd to compute the gradients and verify if the results match with the ones from above

5. Do the same for vector/tensor inputs. Note down the shape of the results at each step and provide proper reasoning for the same

6. The CS231n lecture has a slide on "Patterns in backward flow" explaining about interesting properties of the gradients of specific functions. Demonstrate those properties with simple code snippets.

Activation Functions

Readings:

- CS231n Winter 2016: Lecture 5: Neural Networks Part 2

Exercises:

1. Why is a non-linear activation function important? In particular, the underlying function being learnt by the ML algorithm may totally be unrelated to the activation function being used. So how is this non-linearity going to help?
2. Demonstrate your understanding of different activation functions (atleast 5) by implementing them in Numpy
3. Implement equivalent "backward" functions that would compute the derivative of the activation functions you've implemented above in Numpy
4. Verify the gradients computed by your function with that of PyTorch's autograd

Loss Functions and Gradient Descent

Readings

- CS231n Winter 2016: Lecture 6: Neural Networks Part 3 / Intro to ConvNets
- Mini Batch Gradient Descent (C2W2L01) (YouTube)
- Understanding Mini-Batch Gradient Descent (C2W2L02) (YouTube)
- Exponentially Weighted Averages (C2W2L03) (YouTube)
- An overview of gradient descent optimization algorithms

Exercises

1. Demonstrate your understanding of MSE and Cross Entropy Loss by implementing them in Numpy
2. Implement equivalent "backward" functions that would compute the derivative of the above loss functions in Numpy
3. Verify the gradients computed by your function with that of PyTorch's autograd
4. Learn about Gradient Descent and its below variants:
    a. Momentum
    b. Nesterov
    c. Adagrad
    d. RMSProp
    e. Adam
5. Implement all the above in Numpy
6. How does the "Exponential weighted average" lecture given in the readings relate to some of the variants of Gradient Descent?

Learning Resources

Python

- Official Python tutorial
- Real Python - Tutorials
- Python for everybody
- Fluent Python Book
- Harvard CS50 Python - Beginner course
- virtualenv
    - How to Set Up a Virtual Environment in Python – And Why It's Useful
- Pip
    - What is pip?
    - Pip Python Tutorial for Package Management
- Conda
    - Getting Started with Conda
    - Introduction to Conda
- Poetry
    - Poetry Docs
    - Dependency Management With Python Poetry
    - Why you should use Poetry instead of Pip or Conda for Python Projects
    - How to create a Python package in 2022
- Mypy
    - Static types in Python, oh my(py)!
    - Mypy docs
- Pylint, Black, etc.
    - The Gates of Code Quality — Pylint
    - Improve Your Code with Pylint and Black
    - Static Analyzers in Python
    - The uncompromising code formatter
    - Getting started with Python pre-commit hooks
- Libraries and Tools
    - Jupyter Notebook Introduction
    - Numpy Tutorial - CS231N Stanford
    - Numpy Tutorial
    - Streamlit
    - Parallel Processing for Data Scientists

- 10 mins intro to Pandasx
- Hydra
    - Getting started with Yaml
- Pydantic
- Fast API
- WebDataset - Seeing this in action Refer
    - Why do we need webdataset ?
    - Large Scale Machine Learning with WebDataset - YouTube

Tools

- Tmux
  - A Gentle Introduction to Tmux
  - Complete Tmux tutorial - YouTube
- VSCode
- Docker
  - Docker Tutorial for Beginners - YouTube
  - Play With Docker
  - What is Docker Used For? A Docker Container Tutorial for Beginners
  - Basic Docker commands - Command for Docker output using Rich
- Git - Beginner Introduction
- GitLab CI CD Tutorial for Beginners [Crash Course] YouTube
- DVC
  - Python Data Version Control
  - DVC docs

Audio

- First principles - What's a sound.
- Audio from Scratch
- Let's learn about waveforms
- Digital Audio - The Basics YouTube
    - Digial Audio Fundamentals - Excellent source
    - Basic Audio Concepts
    - Audio Encoding
    - Audio fileformat and Bitrates - Hear the difference
- Audio Signal Processing for Machine Learning - YouTube
- Speech Processing for Machine Learning: Filter banks, Mel-Frequency Cepstral Coefficients (MFCCs) and What's In-Between
- Mel Frequency Cepstral Coefficient (MFCC) tutorial
- FFMPEG
    - A quick guide to using FFmpeg to convert media files
    - 20+ FFmpeg Commands For Beginners
- Audio in Frequency domain
    - Most commonly used featurization techniques
    - Understanding FFTs and Windowing
    - Window Functions in Spectrum Analyzers
    - Implementing FFT from scratch

Machine Learning / Deep Learning / AI

- Math
  - Essence of Linear Algebra - YouTube
  - Essence of Calculus - YouTube
- MIT AI - Introduction YouTube
- A Friendly Introduction to Deep Learning
- Machine Learning Specialization - Andrew Ng
- Deep Learning Specialization - Andrew Ng
- Stanford CS231N - Andrej Karpathy - Lectures 2 to 7 YouTube
- Neural Networks and Deep Learning
- Machine Learning for Engineering and Science Applications - NPTEL YouTube *(Note: The professor is a part of Zoho)*
- Practical Deep Learning for Coders - FastAI, 2022
- Calculus on Computational Graphs: Backpropagation
- The spelled-out intro to neural networks and backpropagation: building micrograd - Andrej Karpathy (YouTube)
- PyTorch
  - PyTorch for Deep Learning - Course YouTube
- PyTorch Lightning
  - PyTorch Lightning Tutorials
  - PyTorch Lightning for Dummies
- CNN
  - Stanford CS231N - Lecture 7
  - Introduction to CNN - NPTEL YouTube
  - A Friendly Introduction to CNN and Image Recognition YouTube
  - NPTEL - ML for Engg and Science Applications
- RNN/LSTM/GRU
  - A Friendly Introduction to RNN YouTube
  - NPTEL - ML for Engg and Science Applications
  - Understanding LSTMs Chris Olah
- Seq2Seq, Attention and Transformers
  - Stanford CS224N: NLP with Deep Learning | Winter 2019 | Lecture 8 – Translation, Seq2Seq, Attention YouTube
  - ai.bythebay.io: Stephen Merity, Attention and Memory in Deep Learning Networks
  - Attention Model - Andrew Ng YouTube

    - Visualizing A Neural Machine Translation Model (Mechanics of Seq2seq Models With Attention)
    - Attention is All You Need YouTube
    - The Illustrated Transformer
- HuggingFace Course
- Neural Network Training
    - A Recipe for Training Neural Networks - Karpathy

Speech Recognition

- Introduction
  - Audio Deep Learning Made Simple: Automatic Speech Recognition (ASR), How it Works
- Machine Learning is Fun Part 6: How to do Speech Recognition with Deep Learning
- Building an End-to-End Speech Recognition Model in PyTorch
- Comparing End-To-End Speech Recognition Architectures in 2021
- Deep Learning for Speech Recognition (Adam Coates, Baidu) - YouTube
- Lecture 12: End-to-End Models for Speech Processing Stanford - YouTube
- Automatic Speech Recognition - An Overview, Preeti Jyothi, IITB YouTube
- Towards an ImageNet Moment for Speech-to-Text
- MIT Automatic Speech Recognition, Rev AI -- YouTube
- Decoding - CTC
  - CMU S18 Lecture 14: Connectionist Temporal Classification (CTC) YouTube
  - An Intuitive Explanation of Connectionist Temporal Classification
  - Beam Search Decoding in CTC-trained Neural Networks
  - Word Beam Search: A CTC Decoding Algorithm
  - Sequence Modeling With CTC
  - Connectionist Temporal Classification, Labelling Unsegmented Sequence Data with RNN | TDLS (YouTube)
- Language Modeling - N grams
  - NLP: Understanding the N-gram language models YouTube
  - Chapter 3 - Natural Language Processing with Dan Jurafsky and Chris Manning, 2012 YouTube
  - Language Models: N-Gram
  - N-Gram Language Models - NPTEL YouTube
  - Ken LM
    - How to Train and Run a Simple Language Model
    - Training an N-gram Language Model and Estimating Sentence Probability
    - KenLM and Arpa Introduction

Speech Enhancement

- What is Speech Enhancement
    - Neural networks based Speech Enhancement
- One of the Deep Learning based approach in solving Speech Enhancement
- TF Frequency bin and different mask representations
- DNN-Based Online Speech Enhancement Using Multitask Learning and Suppression Rule Estimation
- How Noise cancellation Headphones work
- Real-Time Noise Suppression Using Deep Learning