Weekly Mentor + Team Meeting: Every Saturday
5:00-6:00 PM Our github repo: acmucsd-projects/wi23-ai-team-1 (github.com)

Team Meeting Time: 2/18/23 at 5pm in CSE B230
Attendees: Aniket Gupta Arnav Modi Jeffrey Lee Jimmy Ying Steven Shi Vincent Tu Vivian Liu

What have we done so far

- Finished brainstorming project: toxic comment classification
- We're ahead of Team 2 muahahahaha

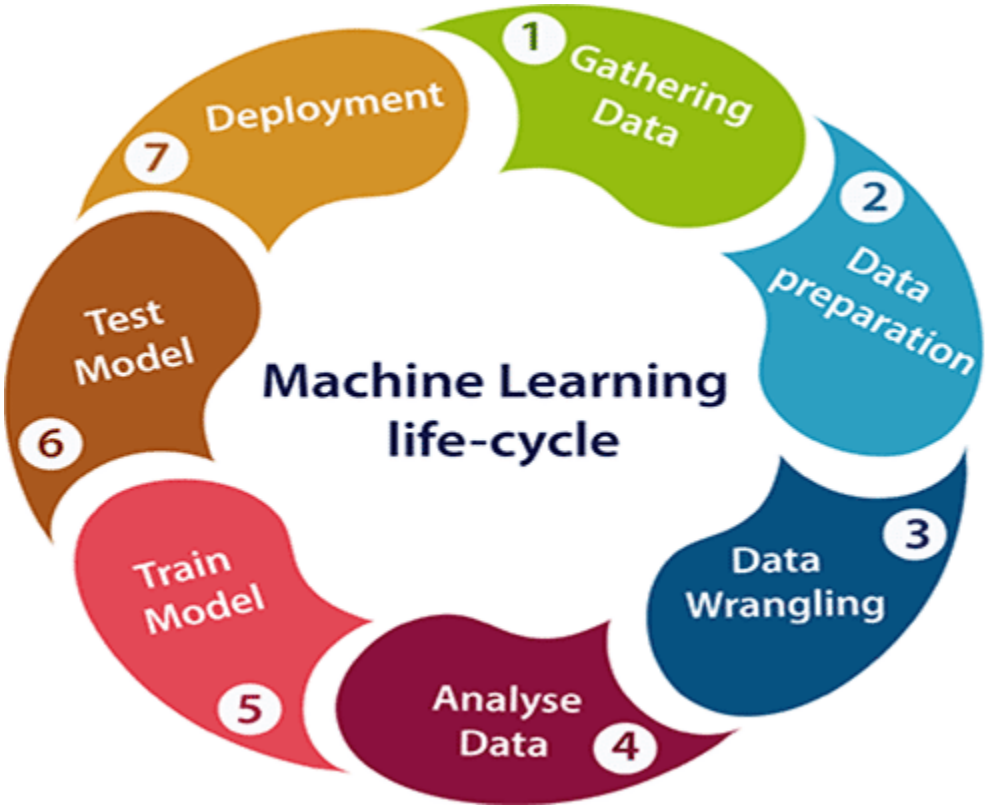
What is the point of this meeting, what are we going to discuss

- Goals for today: logistics, timeline, approach, dividing up work
- Colab or Kaggle Notebook
    - Easier to load data into kaggle
    - Come preinstalled with packages, but might run into problems if versions don't match across environments
        - Can bypass with requirements.txt (list of all the packages and versions)
    - Stick to Kaggle Notebook for now because it's easy to switch from Kaggle to Colab
    - Don't use local laptop (can write code but don't train anything locally)
- To train a good model for this NLP project, you need at least 24 GB. Ideally 32 or 48. At least 12-16 GB GPU memory
- Dataset is around 50MB
- Colab and Kaggle Notebook aren't connected to the repo
    - After a work session, download locally to add it to the repo
- Resources: Vincent, Kaggle (since competition's over and there will be lots of resources)
- Preprocessing
    - Can't feed words into a model, need to turn dataset into numbers
    - Tokenizing: encode words into numbers through vocabulary table (basically a dictionary)
    - Embedding: type of layer in a deep neural network
        - For each number that corresponds to a word, you map it to a vector of n dimensions (n >= 1000) to train the model
    - Tensor: generalization of a matrix so there's more than 2 dimensions, aka multidimensional array
    - All arrays fed into the model have to be the same length

What we did:

1. Make a Kaggle account using your **personal email**
2. Clone GitHub repo to local machine (if you haven't already)
3. Git pull to get the most recent changes
4. Download dataset and put it in your local repo's input folder (extract all and include only the CSVs in input folder). Do not commit the dataset

Timeline:



- **Week till 2/18**
    - Learn pandas, learn numpy, learn matplotlib (don't watch all of them; watch as many as you can)
        - Numpy:
            - https://www.youtube.com/watch?v=QUT1VHiLmmI&ab_channel=freeCodeCamp.org
        - Pandas:
            - https://www.youtube.com/watch?v=vmEHCJofslg&ab_channel=KeithGalli
        - Matplotlib:
            - https://www.youtube.com/watch?v=DAQNHzOcO5A&ab_channel=KeithGalli

    - Work on data wrangling (cleaning the dataset; don't do preprocessing just yet)
- **Week till 2/25**
    - Data exploration (and possibly data preparation/preprocessing)
    - *Check out Competition page "Code" and "Discussion" sections for how to explore, preprocess, and clean the dataset*

Weekly Team Meeting: Every Friday
5:00-6:00 PM Our github repo: acmucsd-projects/wi23-ai-team-1 (github.com)

Team Meeting Time: 2/17/23 at 5pm in CSE Basement
Attendees: **Aniket Gupta** **Arnav Modi** **Jeffrey Lee** **Jimmy Ying** **Steven Shi** **Vincent Tu**

What have we done so far

- We watched the numpy and pandas video

What is the point of this meeting, what are we going to discuss

- Discuss strategies for cleaning data and loading the data in python

What will we do going forward

Weekly Team Meeting:
5:00-6:00 PM Our github repo: acmucsd-projects/wi23-ai-team-1 (github.com)

Team Meeting Time: 2/18/23 at 5pm in CSE Basement
Attendees: **Aniket Gupta** **Jeffrey Lee** **Vincent Tu**

What have we done so far

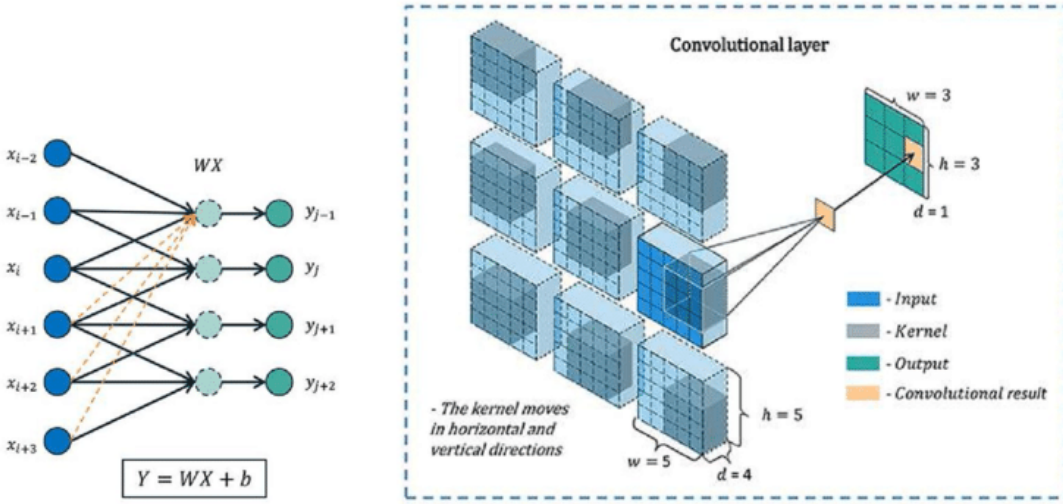
- Added a function to filter out small sentences
- Added a function to filter out all non-ascii characters

What is the point of this meeting, what are we going to discuss

- Filter out numbers and punctuation
- Position embedding to compensate for filtering out punctuation?
- NLTK (natural language toolkit) - library with a bunch of functions for pre-processing
- Code own functions for filtering so that they are as customizable
- Removing other languages: sufficient to just remove non-ascii characters
- Stopwords: filler words in english are used as stop words ("like", "kinda"), used for humans to better read or understand something → these words are wasted text and should be removed
    - Use NLTK.stopwords
    - https://www.geeksforgeeks.org/removing-stop-words-nltk-python/
- Explore the reason for
    - Why there are -1 values in the test label
        - Value of -1 indicates it was not used for scoring?
    - Why there is so much test data
- Must account for class imbalance by checking the amount of data is provided in each class and balance the data between each class
- Filtering Data
    - Removing other languages
    - Tests that are too short
    - Remove special characters (punctuation and new line characters)
    - Remove numbers
    - Remove stopwords

What will we do going forward

- For most collaboration: just write locally → pull and then use and preferred environment (Kaggle)
- Force everything into lowercase first
- Clean dataset
    - Removing other languages
    - Tests that are too short

    - Remove special characters (punctuation and new line characters)
    - Remove numbers
    - Remove stopwords
- After preprocessing and cleaning data → analyse data
- Use Tensorflow for easier way to get into deep learning
    - Tensorflow has all the computation and functions needed
        - Keras is the library that allows you to define layers and networks for deep learning models
        - Layers: collection of nodes
        - 

        - (a) (b)
- Exploration (find patterns and findings to better the models performance)
    - Graph based on
        - Length of the text
        - Which words are most commonly seen
        - Which words are most commonly associated with which kind of toxicity
    - Finding weaknesses and problems
    - Find anything you can exploit
- Preprocessing:
    -
- Modeling (Dense neural network)
    - CNN
        - Recommended to follow this guide using the dataset: https://www.tensorflow.org/text/tutorials/text_classification_rnn
    - BERT:
        - Paper in NLP AI - a novel way to train a transformer (composed of complex layers)
        - A model that can do very well on English text
        - May be used later on along the line
        - Only when moving to a model along the lines of BERT may require using Kaggle
- Experimentation after modeling

Weekly Team Meeting: Every Friday
5:00-6:00 PM Our github repo: acmucsd-projects/wi23-ai-team-1 (github.com)

Team Meeting Time: 2/24/23 at 5pm in CSE Basement
Attendees: **Arnav Modi** **Aniket Gupta** **Steven Shi** **Vivian Liu** **Jeffrey Lee** **Jimmy Ying**

What have we done so far

- Cleaning:
  - Filtering by number of words
  - Filtering non-english chars
  - Trimming whitespace
- Exploration
  - Graphs based on the number of words
  - Frequencies of toxic tags
- Initial steps of tokenization
  - Using tensorflow to create a `vectorize_layer` that separates words in the comment by whitespaces, forces everything to lowercase, and tokenizes to ints

What is the point of this meeting, what are we going to discuss

Trying to optimize our development workflow, looking at how we're collaborating.

More preprocessing stuff:
https://keras.io/api/layers/preprocessing_layers/core_preprocessing_layers/text_vectorization/

What will we do going forward

1) Watch a video on tensorflow and nltk

Weekly Mentor + Team Meeting: Every Saturday
5:00-6:00 PM Our github repo: acmucsd-projects/wi23-ai-team-1 (github.com)

Team Meeting Time: 2/25/23 at 5pm in CSE B230
Attendees: Aniket Gupta Jeffrey Lee Jimmy Ying Steven Shi Vincent Tu Vivian Liu

What have we done so far

- We have cleaned the data
- We have explored the data via plots
- We have begun planning out possible model structures using tensorflow

What is the point of this meeting, what are we going to discuss

- Don't read and rewrite the data in the clean function
- Everything before modeling can be done with a notebook
- Modeling should be done with multiple python files

There is a fast track way to plug in a numpy array into the model, but we will do it the more tedious, complete way, which apparently exists

We ran through the text classification rnn tutorial, mentioned in discord

Notes on pytorch w/ Vincent's old project

- Put data set in a class that inherits from base dataset
- __init__ constructor
- __len__ (how many instances, images, comments, etc.)
- __getItem__ (get an instance selected via an index, perform some kind of augmentation/transformation on it, returns data and label)
- are only functions needed in pytorch dataset, lots of legroom
- Pass in df for constructor, mostly just attribute setting
- Augmentations, tldr: apply transformation to data, ex. Feed rotated image
- This is to improve model performance and generalize, it knows how to handle more data
- We may or may not have augmentations, text augmentation is kinda sus

Generally, no augmentations to text

- Text is sequential, the order really matters

Try to get data working in tensorflow, if that doesn't work, use python

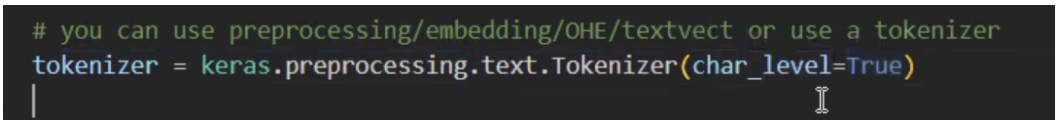
PyTorch	Tensorflow
•	•

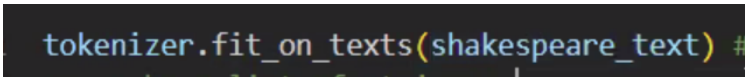
Cleaning = remove unintentional errors and problems with the dataset

Preprocessing: take cleaned data, and convert to compatible format

- Same general steps, but different apis

What will we do going forward

- Convert our pandas df, preprocessing →
- Preprocessing & model
- **Decide between pytorch and tensorflow, make a dataset in both**
- Consider augmentation and feature engineering, probably won't be too useful, but fun to know
- Tokenize / text vectorization before preprocessing
  - 

  - 

- **Tokenizer maps keys (word) to values (integer)**

Weekly Mentor + Team Meeting: Every Saturday
5:00-6:00 PM Our github repo: acmucsd-projects/wi23-ai-team-1 (github.com)

Team Meeting Time: 2/25/23 at 5pm in CSE B230
Attendees: Aniket Gupta Jimmy Ying Steven Shi Vincent Tu Vivian Liu

What have we done so far

- We made an initial model
    - Added vectorization layer
    - Added embedding layer
    - Added bidirectional layer?
    - Added 2 dense layers?
- Immense progress

What is the point of this meeting, what are we going to discuss

- Discuss what the model does
- Average pooling
    - Reduces dimensions of the output of the embedding layer
    - Prevents overfitting

What will we do going forward

- Early stopping
    - Add validation dataset
    - Stop if the accuracy goes down or doesn't go up for 2-5 epochs
- Random seeds

Weekly Mentor + Team Meeting: Every Saturday
5:00-6:00 PM Our github repo: acmucsd-projects/wi23-ai-team-1 (github.com)

Team Meeting Time: 2/25/23 at 5pm in CSE B230
Attendees: Aniket Gupta, Jimmy Ying, Steven Shi, Vincent Tu, Vivian Liu, Arnav Modi

What have we done so far

- We added another LSTM layer
- We added random seeds so our results are consistent
- Reorganized the github

What is the point of this meeting, what are we going to discuss

- We added regularization layer
    - Dense layer
- Figuring out why our data is overfitting

What will we do going forward

- Attempt to properly implement regularization - Vivian
- Expand model to account for all 6 toxicity labels
- Work on improving our model
- Shuffle the dataset before splitting
    - Stratified shuffle
    - Sklearn stratified shuffle split - Vivian
- Fix the random seed - aniket
- Figure out why training takes so long
    - Maximize the batch size (set to like 64) - steven
    - Cut down the embedding layer: len(encoder.get_vocabulary()) - jimmy
    - GOOGLE HOW TO CUT DOWN TRAINING - jimmy
    - Graph the frequency of words - jimmy
        - Figure out how to narrow down vocabulary
- Use correct metrics (parameter in model.compile function) or fix the class imbalance problem - arnav
- ~~Figure out the workflow (how we gonna collaborate) - done~~
    - A folder for each person
    - Naming conventions
- Double check that the pre-processed format of the dataset is correct when going into the model
- Figure out how to split the dataset

    - Find a good & effective split so that we have a good training set and a good validation dataset

● Someone make a submission to the competition - steven

- Don’t use the test dataset, use the validation instead
- Fix these weird problems before we start experimenting
- Standardize our notebooks (implement a format)
- Notebook convention:
    - Setup
    - Cleaning (already done)
    - Preprocessing
    - Model
    - Training
- add 5 fold cross validation with sklearn stratifiedshufflesplit
    - More robust way to validate performance improvement

Weekly Team Meeting: Every Friday

5:00-6:00 PM Our github repo: acmucsd-projects/wi23-ai-team-1 (github.com)

Team Meeting Time: 3/10/23 at 5pm in CSE Basement
Attendees: **Arnav Modi** **Aniket Gupta** **Steven Shi** **Vivian Liu** **Jeffrey Lee** **Jimmy Ying**

What have we done so far

- Increased batch size to 64
- Limited the vocabulary size to 5000 and limited the length of each phrase to 200 words
- Tried using imblearn to fix imbalance (did not work), also tried class weight but the accuracy was decreasing on increasing the weight when the target value was 0

What is the point of this meeting, what are we going to discuss

- Merged all our different versions of the notebook file

What will we do going forward

- Now that we're finished ironing out all the issues, focus on improving the model
- We need to improve the entire pipeline, improve all the steps, not just the model
- Add more useful layers, test around (One Person:) (EVERYONE)
    - Optimizer, Dropout, Dense, Regularizer, Hidden Dimensions, "There's more to it"
    - Experiment with LSTM/GRU layers - Aniket
    - Dropout, Dense, CONV1D - Arnav
    - Dropout (experiment with the values and experiment with sandwiching between layers) - JIMMy
    - Regularizer and hidden dimension layers - steven
- //Look for specifics, all caps, punctuation, heuristic (hard coding a checker)
- Fix class imbalance issue (Arnav does all the work, everyone else "looks into it")
- Get model to predict in all six categories (Steven and Aniket do all the work, everyone else "looks into it")
- Figure out a way to monitor/track the progress
    - Maybe create a spreadsheet
    - Keep track of what works and what doesn't work

Weekly Mentor + Team Meeting: Every Saturday
5:00-6:00 PM Our github repo: acmucsd-projects/wi23-ai-team-1 (github.com)

Team Meeting Time: 3/11/23 at 5pm in CSE Basement
Attendees: **Aniket Gupta** **Jimmy Ying** **Vincent Tu** **Vivian Liu** **Arnav Modi**

What have we done so far

- Nothing since yesterday

What is the point of this meeting, what are we going to discuss

- Made immense progress
- Experimentation Spreedsheet
- Arnav and Aniket: changed modeling.ipynb to incorporate all 6 labels, changed the final dense layer from 1 to 6, kaggle comp submission now gives us 68% (up from 53%) woohoo!!!!
- Jimmy: Experimenting with dropout layer (with the old one-label model), recording results in spreadsheet

What will we do going forward

- Idk figure stuff out
- Look into the following metrics: recall, precision, h1 (harmonic mean of recall and precision) instead of purely going off accuracy - Aniket
- Sanity check every step of the pipeline
- Check the data that is going into the model: how many data points, shape of tensors, indices, datatype, everything is vectorized/encoded properly, if it is in the vocabulary that it makes sense, etc.
- Make sure the dataset actually makes sense, generate a few outputs from it
- Make sure we have correct activation function in the output layer, pass in some dummy data
- If we have 10 metrics, remove most of them and see what exactly is wrong. Strip away the unnecessary details
- STRATIFY THE VAL DATA SET

Weekly Mentor + Team Meeting: Every Saturday
6:00-7:00 PM Our github repo: acmucsd-projects/wi23-ai-team-1 (github.com)

Team Meeting Time: 3/17/23 at 6pm in CSE Basement
Attendees: **Aniket Gupta** **Vincent Tu** **Steven Shi** **(10 minutes)**

What have we done so far

- https://docs.google.com/spreadsheets/d/1qszCECyEmNuFSlsMiOAVaRIbhtOy3UWqbqm6ExHmnQU/edit?usp=sharing
- Validation accuracy is skewed because model just guesses 0 all the time for certain labels

What is the point of this meeting, what are we going to discuss

- Immense progress
- Overfitting very fast
    - Lots of regularization
    - Batch norm
    - More dropout layers
    - Caused by imbalance
- Overfit on one batch to test if model can learn
- VERY USEFUL BLOG - http://karpathy.github.io/2019/04/25/recipe/

What will we do going forward

- Same as earlier
- Look into the following metrics: recall, precision, h1 (harmonic mean of recall and precision) instead of purely going off accuracy - Aniket
- Add more useful layers, test around (One Person:) (EVERYONE)
    - Optimizer, Dropout, Dense, Regularizer, Hidden Dimensions, “There’s more to it”
    - Experiment with LSTM/GRU layers - Aniket
    - Dropout, Dense, CONV1D - Arnav
    - Dropout (experiment with the values and experiment with sandwiching between layers) - JIMMy
    - Regularizer and hidden dimension layers - steven

Weekly Mentor + Team Meeting: Every Saturday

6:00-7:00 PM Our github repo: acmucsd-projects/wi23-ai-team-1 (github.com)

Team Meeting Time: 3/18/23 at 6pm in CSE Basement
Attendees: **Aniket Gupta Arnav Modi Jimmy Ying Vincent Tu Vivian Liu**

What have we done so far

- Not much progress in the past week

What is the point of this meeting, what are we going to discuss

- Read through this very useful blog - http://karpathy.github.io/2019/04/25/recipe/
- Getting higher accuracy for toxic category

What will we do going forward

- Fix imbalance
- Will be meeting over break

Emergency Meeting

6:00-7:00 PM Our github repo: acmucsd-projects/wi23-ai-team-1 (github.com)

Team Meeting Time: 4/3 at 6pm at Geisel
Attendees: **Aniket Gupta Arnav Modi Jimmy YingSteven Shi Vincent Tu Vivian Liu (10 minutes)**

What have we done so far

- Immense progress
- We have a minimum viable product

What is the point of this meeting, what are we going to discuss

- Figure our how to use streamlit to deploy our app
- Save the model
- Continue optimizing

What will we do going forward

- Work on the slides for our presentation
- Get the streamlit working