COPS Summer of Code 2025

Intelligence Guild

Club Of Programmers, IIT (BHU) Varanasi

NLP Track: Sequence Modeling

2 – 8 June 2025

Official IG Website: https://cops-iitbhu.github.io/IG-website/

All deadlines are strict. No extensions will be granted.

Introduction

COPS Summer of Code (CSOC) is a flagship initiative under the Club Of Programmers, IIT (BHU) Varanasi, with all verticals contributing through focused tracks. This document embarks the journey of deep learning and contains the contents of ANN.

Modules will be released from time to time. **Adhere strictly to deadlines**. Submissions will be evaluated on approach, technical correctness, and clarity. The most technically accurate solution may not necessarily be the one chosen; clarity of thought and a well-reasoned approach will be valued more.

Communities

All communication for the programme will be conducted strictly via Discord. Do not reach out through other channels. Resources and updates will be posted on Github, and all notifications will be made via Discord.

Final Report

A concise report may be submitted along with your final assignment. While **not mandatory**, it may strengthen your overall evaluation. Reports must be written in LaTeX and submitted in PDF format only. We are not interested in surface-level descriptions — focus strictly on your analysis, approach, and reasoning. The report itself constitutes the final assignment. No additional files are to be submitted. Refer to the Assignment section for details.

Contact Details

In case of any doubts, clarifications, or guidance, you can contact one of us. We request that you stick to Discord as the preferred mode of communication for all the questions that you have as it will also benefit others. However, you can reach out to us through other means in case we fail to respond on Discord.

- Tejbir Panghal - 9034705165
- Sakshi Kumar - 8073247266

Resources

In this module, we'll dive into the world of **Recurrent Neural Networks (RNNs)**, **Long Short-Term Memory networks (LSTMs)**, **Gated Recurrent Units (GRUs)**, and **Word Embeddings**. These are crucial concepts for processing sequential data in NLP.

Recurrent Neural Networks (RNNs), LSTMs, and GRUs

This section will guide you through the fundamental concepts of sequence models.

- If you're looking for clear, intuitive explanations, check out**StatQuest with Josh Starmer**:
  - For RNNs: Recurrent Neural Networks (RNNs), Clearly Explained!!!
  - For LSTMs: Long Short-Term Memory (LSTM), Clearly Explained
- For a visual and step-by-step breakdown of both LSTM and GRU: Illustrated Guide to LSTM's and GRU's: A step by step explanation by The AI Hacker.
- To get hands-on with implementing these in PyTorch: PyTorch Tutorial - RNN & LSTM & GRU - Recurrent Neural Nets by Patrick Loeber.
- For a deeper understanding by building an RNN from scratch in Python: RNN From Scratch In Python by Dataquest.

If you prefer reading over videos, these resources are excellent:

- A comprehensive article covering RNNs, LSTMs, and GRUs: A Journey Through RNN, LSTM, GRU and beyond.
- **Colah's blog** is highly recommended for its in-depth and intuitive explanations, especially for LSTMs: Understanding LSTMs.

Word Embeddings

Understanding how words are represented numerically is key.

- For the basics of text preprocessing and early embedding techniques:
  - Text preprocessing steps: Understanding the Essentials: NLP Text Preprocessing Steps.
  - NLP Zero to Hero (Part 1), covering Bag of Words, TF-IDF, and an introduction to Word2Vec: Introduction, BoW, TF-IDF, Word2Vec.
- **Krish Naik** offers great conceptual videos:
  - Bag of Words intuition: Natural Language Processing—Bag Of Words Intuition.
  - TF-IDF intuition: Natural Language Processing—TF-IDF Intuition— Text Preprocessing.

  - An introduction to Word Embeddings: Word Embedding - Natural Language Processing— Deep Learning.

- For a deep dive into Word2Vec:
  - An excellent breakdown of skip-gram and CBOW: Word2Vec Explained – by Lilian Weng.
  - A video explanation of Word2Vec: Word Embedding and Word2Vec, Clearly Explained!!! by StatQuest with Josh Starmer.
  - A highly visual and intuitive explanation: The Illustrated Word2Vec – from Jay Alammar.

Extras

For those eager to explore further:

- A systematic empirical analysis of LSTM variants: LSTM: A Search Space Odyssey.
- A classic and insightful blog post by Andrej Karpathy: The Unreasonable Effectiveness of Recurrent Neural Networks.
- Stanford's renowned NLP course material (Week 1 is highly relevant): Stanford 224n course.
- The original Word2Vec paper: Efficient Estimation of Word Representations in Vector Space (Mikolov et al., 2013).
- For understanding more advanced RNN architectures:
  - Bidirectional RNNs (BiLSTM, BiGRU): Bidirectional RNN — BiLSTM — Bidirectional LSTM — Bidirectional GRU by CampusX.
  - Deep/Stacked RNNs, LSTMs, and GRUs: Deep RNNs — Stacked RNNs — Stacked LSTMs — Stacked GRUs by CampusX.
- A personal favorite NLP course in blog form, covering a wide range of topics: https://lena-voita.github.io/nlp$_course.html$.

Assignment: Binary Sentiment Classification

Objective:

Build a sequence-based model (RNN/LSTM/GRU) to predict sentiment polarity (positive or negative) from product review text. Go beyond detecting keywords — the model should learn subtle tone, sarcasm, and emotional cues, encouraging thoughtful model design.

Dataset Summary:

- **text**: Full review written by a user.
- **title**: Title of the review
- **polarity**:
  - 1 → Negative
  - 2 → Positive

Dataset link: Amazon Reviews Dataset

Task Description:

- Preprocess and tokenize the text (cleaning, padding, truncating).
- Use GloVe or Word2Vec embeddings (trainable or pre-trained).
- Train a sequence model (RNN, LSTM and GRU or their variants preferred) for binary classification.
- Evaluate using accuracy, F1-score, and confusion matrix.
- Analyze errors — especially false positives/negatives where the language is ambiguous or sarcastic.

Bonus : Minimal Clue Challenge:

- Pick 5–10 short, ambiguous reviews that failed on the trained model and answer:
  - Why might your model misclassify this?
  - What clues could a human pick up that your model can't?
  - How would you fix this?

Stretch Goals:

- Compare the lengths of reviews that the RNN model predicts correctly versus those predicted by the LSTM/GRU models.
- Hence, analyze how review length or rare words affect performance.

Submission Guidelines

- Create a GitHub repository named `<roll_number>-CSOC-IG` (e.g., `23014019-CSOC-IG`)
- Repository organization:
  - A folder named "Sequence Modelling Basics" containing all source code implementations
  - The final report in PDF format, authored using LaTeX

Everything must be in the github repo itself.

- Submit the repository link via the provided Google Form here
- **Note:** The report constitutes the primary assignment submission. No additional files are required
- **Deadlines are strict and will not be extended**

Final Remarks

Ensure that your submission reflects a clear understanding of the concepts and methodologies applied. Focus on the analytical aspects and the rationale behind your implementations. We look forward to your insightful contributions.

Adios*, and keep learning!*