TOC Project Summary

Abhinav Nakarmi and Kuber Shahi

Project Topic: *Neural Network for Synthesizing Deterministic Finite Automata* (link)

Project Description: Generate a Deterministic Finite Automata(DFA) with the help a Recurrent Neural Network. The problem statement can be formulated as follows:

- **Input**: a Regular Language L to generate the list of tuples (w, ans) for training where ans = 1 if $w \in L$, and ans = 0 if $w \notin L$.

  For example: for language L = { $w \in \sum^*$ | $\sum$= { a, b} and w has no same neighboring characters}, then list = [(abab, 1), (abaa, 0),...]

- **Output**: DFA M such that L(M) = L

Project Intended Goals: Before we started, the primary goal of our project was to understand, explain and implement a working model of a neural network that can synthesize deterministic finite automata for a given formal language. In detail, our goals were:

- To understand how neural networks (specifically RNNs) are used in building DFAs.
- To understand the novel architecture based on RNN proposed by the paper in synthesizing DFAs correctly and quickly.
- To implement the proposed RNN model and achieve the results obtained by the paper
- To get an idea about different RNN models used in synthesizing DFAs mentioned in the paper.

Achieved Goals: At the end of our project, we:

- Understood how Neural Networks (NNs) are able to approximate non-linear functions.
- Understood the core ideas of Recurrent Neural Networks (RNNs), its comparison with NNs, and the problems it can solve (Sequence Modelling).
- Understood and appreciated the proposed RNN to synthesize DFAs, and problems (vanishing/exploding gradient) that the underlying method has.
- Went through the implementation in-detail, and further tested it on our languages.

Implementation Details: The implementation of the proposed RNN and the python files for generating languages have been commented and uploaded on the Github (link).