Ben Waldman
Sophia Madejski

Project Proposal

Problems to be Considered

In World War 2, the German military used a cipher machine called the Enigma to encode secret military information. The machine encodes individual letters deterministically based on user-controlled settings and the previous letters encoded in the same message. The code was eventually broken by the Allies and played a vital part in their eventual victory in the war via a combination of manual cryptanalysis and very early computers. Our goal is to train a neural network that, when provided settings and an encrypted message, can successfully decode enigma encrypted messages. A neural network is uniquely well-suited for this task since each layer of the network can simulate a step in the encryption of messages to produce a final output. Furthermore, dimensionality reduction techniques like PCA and auto-encoding are especially suitable since a perfect model would be able to reduce the data down to one feature: the decrypted string. While the Allies efforts were extremely impressive for their time, we know that modern computational power can do much better.

Datasets

Initially we wanted to train on existing encrypted Enigma text data from actual WWII communications, but we weren't able to find a dataset for this. Instead, we plan on generating encoded data with an enigma simulator. Specifically, we intend to use this repository. In order to have enough text data to successfully train our model, we plan on generating two data sets of 10,000 words from **(a),** coherent and complete text sources: wikipedia pages, and **(b),** nonsense words that will be randomly generated. Since enigma encryptions are not dependent on grammatical structure or coherence of text data, there theoretically should be no difference between these two data sets, but we want to see if this will actually be reflected in our results. We plan to use the encrypted text and each individual setting for the machine as features and the decrypted text as labels.

Potential Concerns

Our main concern right now is about compute. If we have compute issues we were thinking of running our jobs on the peanut cluster. An additional concern is about the size of our data (is 10,000 words enough?). Additionally, does our project adhere closely enough to what we have done in class to be considered viable?

Resources

Learning the Enigma with Recurrent Neural Networks (this resource addresses a similar problem but has a different approach from what we plan to do)
Machine - crypto_enigma.machine — crypto-enigma 0.2.1 documentation