Code-breaking Neural Networks

Tristan Hodgson

December 15, 2025

1 Motivation

During this project, I investigate the ability of neural networks to learn to decrypt cipher text. While this is is not possible for general encryption schemes (e.g. AES and RSA), if we restrict ourselves to less cryptographically secure ciphers, it may be possible for neural networks to learn the underlying transformation from examples alone.

2 Proof of Concept

As a first proof of concept, we focus on the Caesar cipher, this is because it has a finite space of possible keys so we can use a classification model to predict the shift value directly, rather than requiring a more complex seq2seq model. This serves as a useful test of feasibility before we move onto more complex tasks, however it is not generalisable to other ciphers as they have much larger key spaces: even a substitution cipher has 26! possible keys!

In this proof of concept, we use a basic 1D CNN architecture to classify the shift value used in a Caesar cipher. The model is trained on the same Guttenberg text dataset used in the main project (see). The text is preprocessed by applying a random Caesar shift to each sequence, and the model is trained on pairs of $(\text{ciphertext}, \text{shift value})$.

Even with very little data and training time we find that the model achieves a very high level of accuracy (100% to 4s.f. after one epoch) on the validation data set. This is a good sign for the feasibility of using neural networks to learn to decrypt ciphers, and suggests that more complex architectures could be used to tackle more complex ciphers in future work.

3 Data

All data is gathered from the Project Guttenberg[1] corpus of public domain books. Text is preprocessed by converting to lowercase and removing all non-alphabetic characters. Aside from the proof of concept described above, we train using pairs of $(\text{ciphertext}, \text{plaintext})$ generated by applying a cipher to the plaintext. The ciphers we consider in this project are the Caesar cipher, the Substitution cipher and finally the Enigma machine.

4 Model

We experiment with two seq2seq architectures that have been used in the literature: a 1D Convolutional Neural Network (CNN) and a recurrent network based on Long Short-Term Memory (LSTM) units.

The 1D CNN architecture is motivated by its success in ciphertext classification and pattern detection tasks [2]. Convolutional filters can act as detectors for short-range structure, similar to n-gram methods traditionally used in classical cryptanalysis. For simple ciphers such as Caesar and substitution ciphers, this inductive bias is often sufficient to uncover key-dependent patterns [3]. Our CNN uses filters of sizes 2,3,5,7 chosen to capture common n-gram length in English text.

The second architecture is an LSTM-based recurrent neural network[4]. This model is a replication of a model demonstrated by Greydanus to learn decryption of ciphers including Enigma. What is new in our work is the way in which we train this model on a large corpus of natural language text, rather than providing the key alongside the ciphertext. Unlike CNN models, the LSTM is stateful, allowing it to capture dependencies from across the cipher text. This is of particular importance for Enigma messages due to the relatively high complexity of the underlying transformation.

Training is supervised: the model receives ciphertext sequences and must predict the corresponding plaintext tokens. We train on pairs (ciphertext, plaintext), feeding character embeddings into the model and predicting the next plaintext character at each timestep until we predict the end of string character.

5 Results

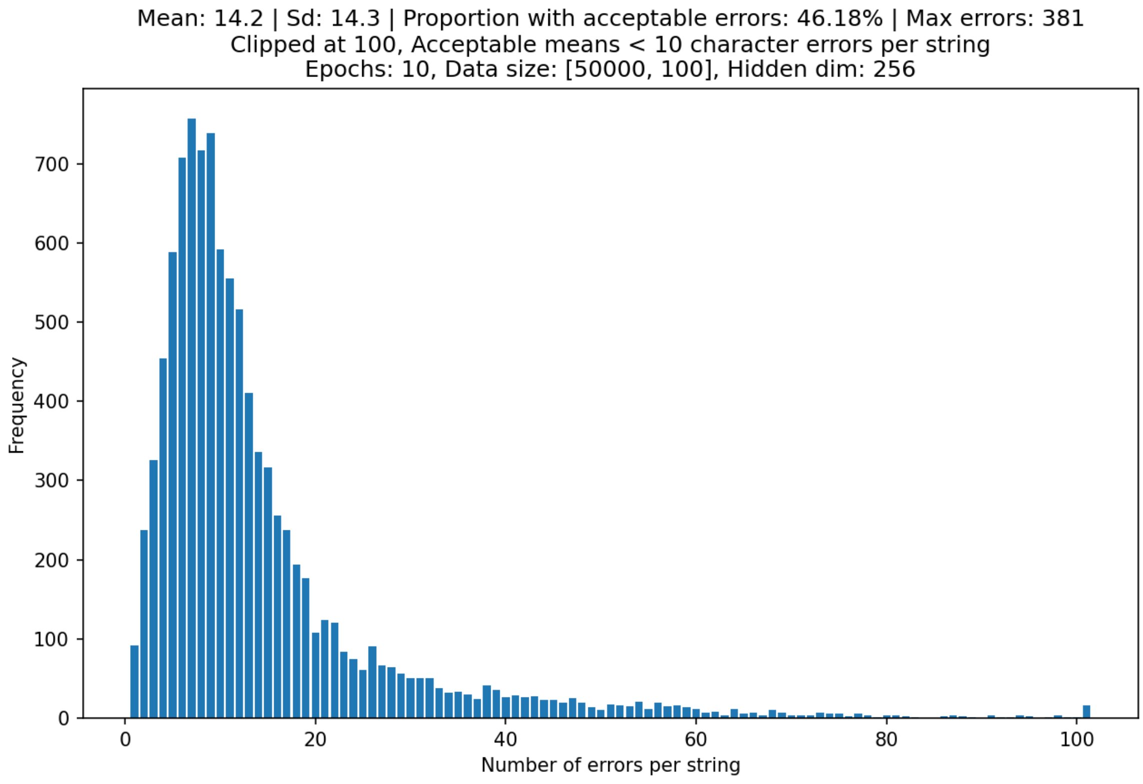

Figure 1: Accuracy of the LSTM model for the Caesar Cipher

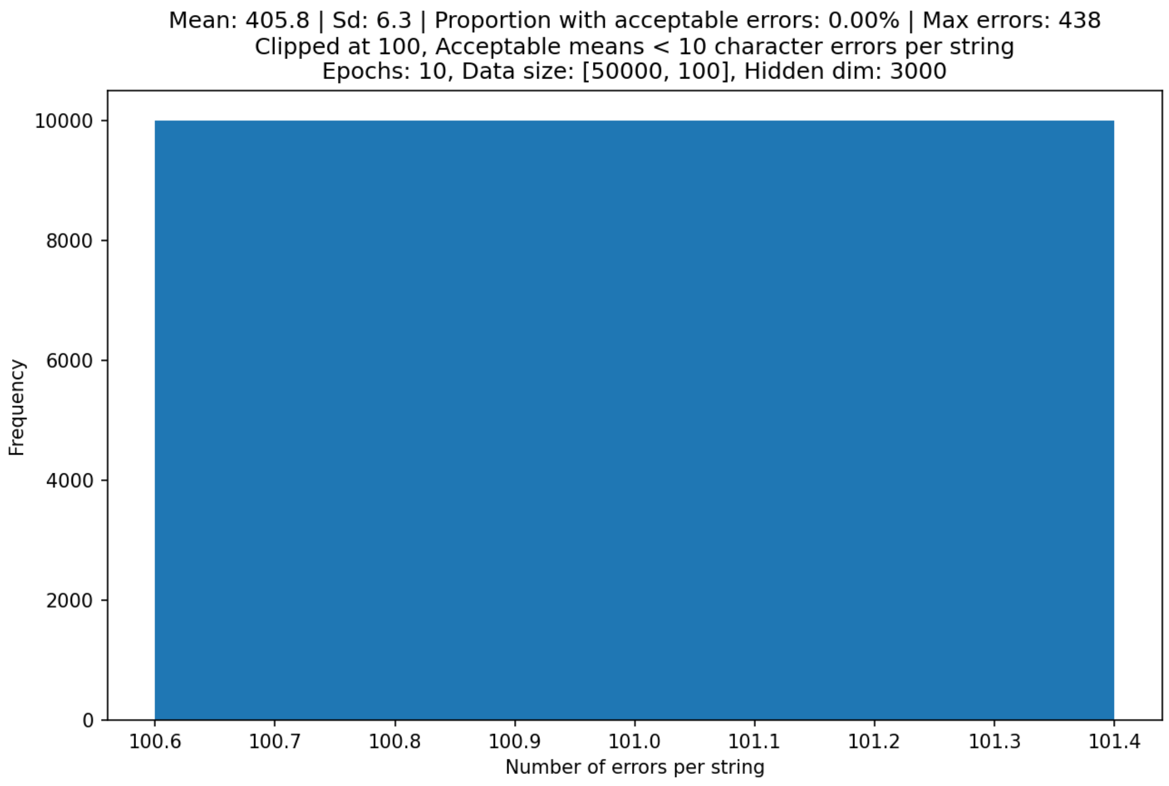

Figure 2: Accuracy of the LSTM model for the random (true) Enigma

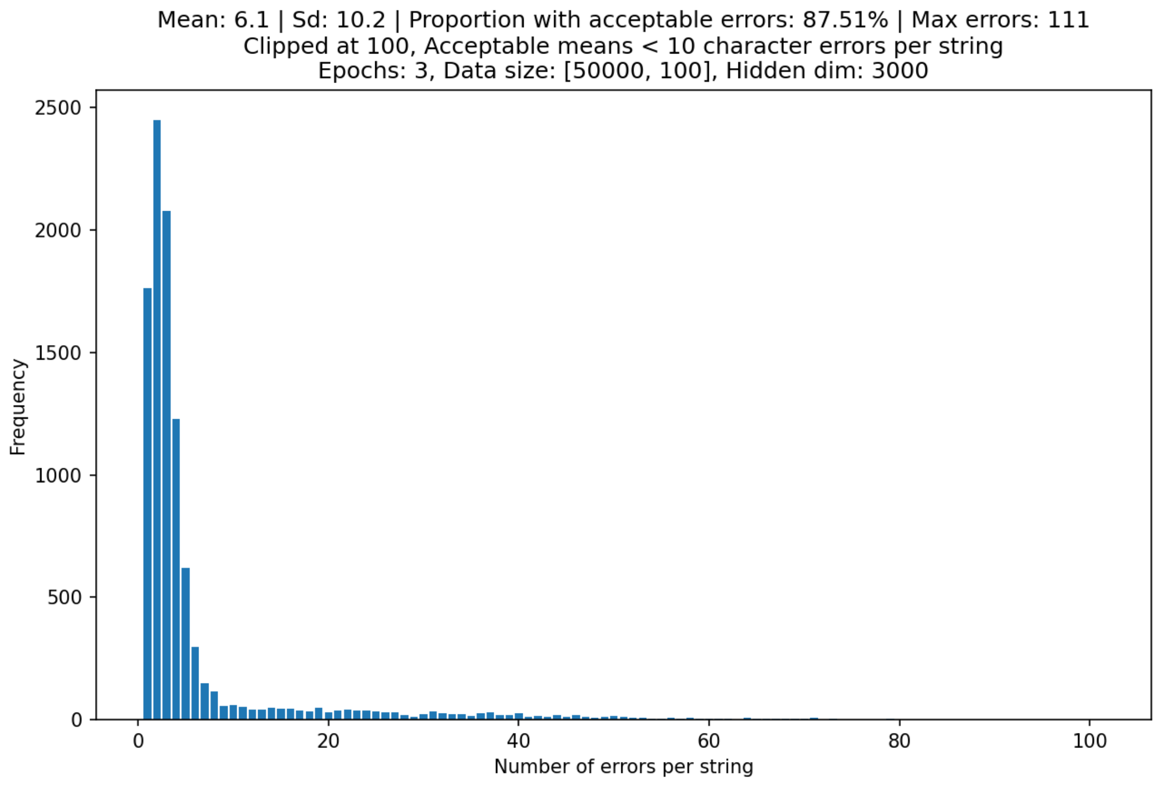

Figure 3: Accuracy of the LSTM model for the fixed Enigma

6 Future Work

Several extensions could build on this proof of concept:

- Trying adding a fixed beginning to the plaintext to see if the model can learn to decrypt general enigma messages given this fixed encrypted header (similar to adding the key as in [4])
- Trying different languages (e.g. German) to see if the results are language dependent
- Trying mixed ciphers, i.e. randomly switching between two ciphers for each character in the plaintext

These extensions would help determine how far neural approaches can be pushed into genuine cipher analysis and more sophisticated symbolic tasks.

References

[1] Project gutenberg. https://www.gutenberg.org/. Accessed 2025-11-22.

[2] Ezat Ahmadzadeh, Hyunil Kim, Ongee Jeong, Namki Kim, and Inkyu Moon. A deep bidirectional lstm-gru network model for automated ciphertext classification. *IEEE Access*, 10:3228–3240, 2022.

[3] Riccardo Focardi and Flaminia L. Luccio. Neural cryptanalysis of classical ciphers. In *Proceedings of the 3rd Italian Conference on Cybersecurity*, 2018.

[4] Sam Greydanus. Learning the enigma with recurrent neural networks. *arXiv preprint arXiv:1708.07576*, 2017.