

Data

Data

Motivation

Goal: Develop a neural network that successfully learns to translate encrypted text to decrypted text using substitution ciphers and the Enigma cipher.

Figure 1. *Staphylococcus aureus* strains.

- Test how effectively neural networks adapt to strict monitoring tasks
- Highlight the developments in science over the past 80 years
- Include an interesting picture

Below: Picture of a white Ensigna machine
Top Right: Onset detecting that could automatically take through an Ensigna machine
Right: Peter being worked on the Ensigna machine, which could automatically keep the Ensigna code

Approach

The Encoder-Decoder based model



- Count the letter frequencies in an encoded sequence
- Match letters to the most common English word frequencies

- Create an encoder-decoder based neural network using PyTorch - based on the outline to the right
- Utilize byte-level encoding
- Stack encoder-decoder blocks to improve performance

- Method #3: Fine-tuned Google BERT
- Load Google BERT model from Huggingface

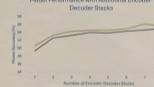
- Load Google Analytics and from Haggglase

- Free base model on cipher data

Building up the Model

Does a larger model improve results?

Model Performance with Additional Encoder.



- Multiple users trained on 50% of available data for fewer's sake
- Partitioning could potentially be achieved & accompanied by an increase in training data
- Output evaluation determining whether to increasing model size
- More stacks

Takeaway

Simple Neural Networks Fail to Adequately Decipher Messages

Challenges
<ul style="list-style-type: none"> 1. Testing time and GPU access significantly limited the complexity of models 2. Working across a very diverse data set of text 3. Very specific output specifications <ul style="list-style-type: none"> • Needed to line up with the input channels to character • Exact iterations 4. Mozilla did not flow smoothly to the next phase

Takeways
<p>Neurot metabolites struggle to adapt to higher levels</p> <p>Problems that occur more complex to humans patients for some using neurot metabolites</p> <p>Free-living struggles when a very specific output is required</p> <p>Increasingly in-vivo size generally improve performance, but with diminishing returns</p>

**Future
institutions**

Breaking Ciphers with Neural Networks

Nick Cichoski



Motivation



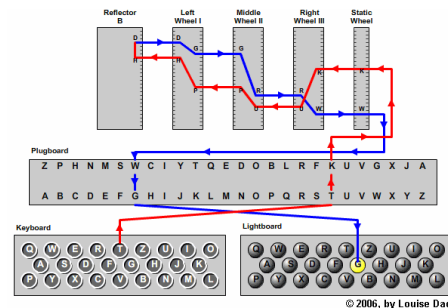
Can a neural network effectively adapt to complex logical tasks?

8

Goal: Develop a neural network that successfully learns to translate encrypted text to decrypted text using substitution ciphers and the Enigma cipher.

Motivation:

- Test how effectively neural networks adapt to strict reasoning tasks
- Highlight the developments in computer science over the past 80 years
- Solve an interesting problem



Above: Picture of a WWII Enigma machine
Top Right: Chart depicting the route electricity takes through an Enigma machine.
Right: Alan Turing working on the Bombe machine, which would eventually break the Enigma code



Data



Data Processing:

- Downloaded from Project Gutenberg
- Every ten lines of text were joined
- Text lines were shuffled
- Text was fed into an enigma machine or a substitution cipher

Data Sources:

- Shakespeare's works
- The Bible
- War and Peace
- Ulysses

Data Statistics:

- Lines: 27,529
- Chars per line: 492
- Words per Line: 94

Substitution Cipher Example

Original Text: as he sat upon the mount of olives the disciples came unto him privately saying tell us when shall these things be and what shall be the sign of thy coming and of the end of the world

Cipher: pz do zpu jqsc udo rsjcu sx slfeoz udo afztfqloz tpro jcus dfr qvfepuolh zphfcy uoll jz idoc zdp11 udozo udfcyz ko pca idpu zdp11 ko udo zfyx sx udh tsrfcy pca sx udo oca sx udo isvla

Enigma Cipher Example

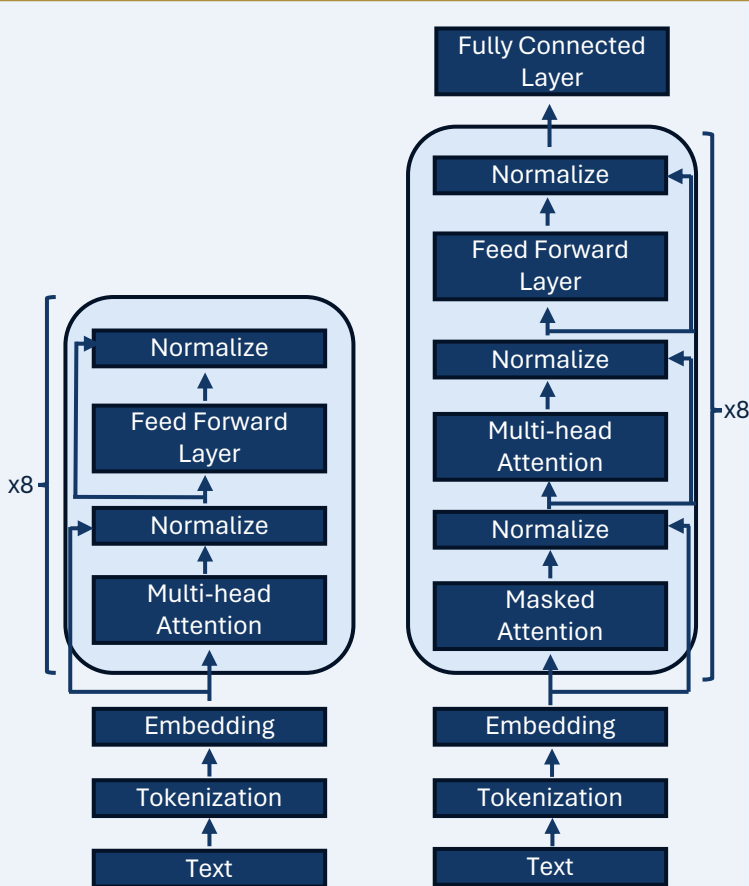
Original Text: stand ho give the word ho and stand what now lucilius is cassius near he is at hand

Cipher:

MBVUYRIDZNYTGGHDPJOBODSURIANBDVUPYIH
ZTOCFGNGHODFCZIXFOMWKVGRNLQKXWJOJL

Approach





Method #1 (baseline): Letter Frequency

- Count the letter frequencies in an encoded sequence
- Match letters to the most common English word frequencies

Method #2: Encoder-Decoder Model

- Create an encoder-decoder-based neural network using PyTorch – based on the outline to the right
- Utilize byte-level encoding
- Stack encoder-decoder blocks to improve performance

Method #3: Fine-tuned Google Byt5

- Load Google/byt5-small from Huggingface
- Fine tune model on cipher data

Results



Accuracy by Model

Model	Substitution Cipher	Enigma Cipher
Baseline	4.54%	4.09%
Encoder-Decoder	63.11%	63.12%
Byt5-Small	NA	4.45%

Takeaways:

- Small encoder-decoder architectures fail to fully solve the ciphers
- The model adapts to substitution and enigma cyphers equally well
- Byt5 requires significant fine-tuning to adapt it to the very specific task

Substitution Cipher

Text: and rack thee in their fancies enter mariana and isabella welcome

Encoding: avs waxj npyy iv npyiw lavxiyq yvnyw rawiava avs iqafyhha myhxkry

Generation: eyx.vqx.q:.x.vqx.q:.xeox.vqx.npqxeox.vqx.

Enigma Cipher

Text: any moment get angrythat at his slightest inattention

Encoding: UBSPCFKAXFKDTMGNKUZMLZPYPHMFGWWBJXNOBRLOASVOBFBCHCRGE

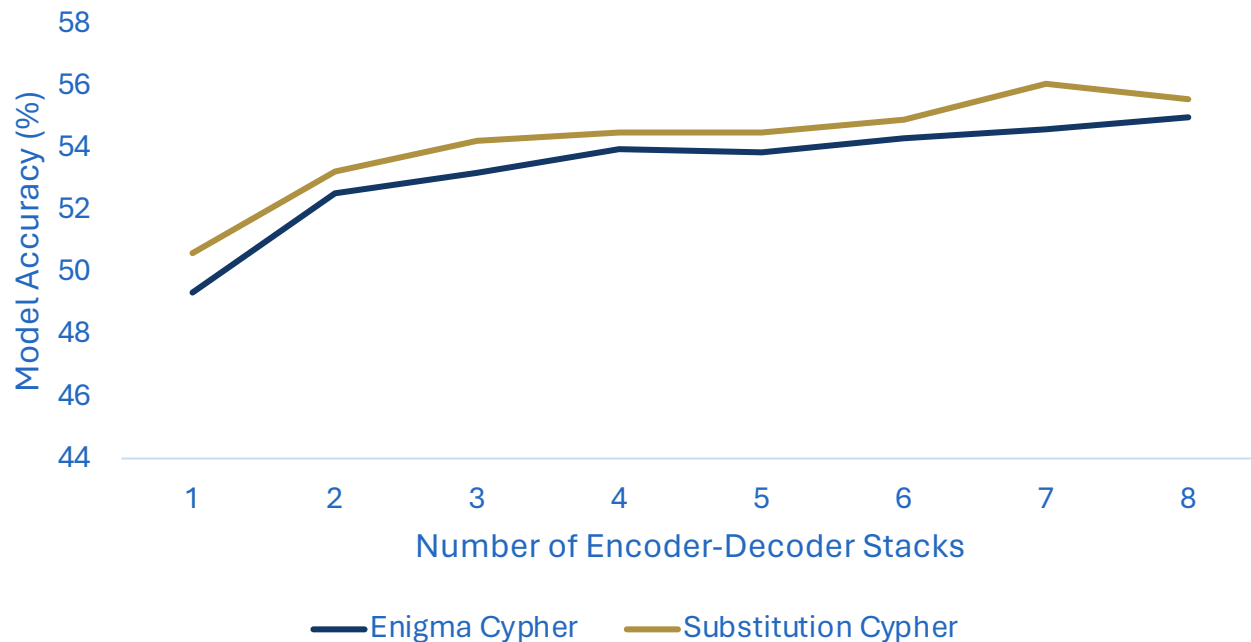
Generation: nvdgqnuregrengs,v.qguoqgrengs,v.qguoqgrengs,

Building up the Model



Does a larger model improve results?

Model Performance with Additional Encoder-Decoder Stacks



Notes:

- Models were tested on 10% of available data for time's sake
- Flatlining could potentially be reduced if accompanied by an increase in training data
- Output indicates diminishing returns to increasing model size
- More stacks generally results in better results

Takeaway



Challenges

- Training time and GPU access significantly limited the complexity of models
- Working across a very diverse data set of text
- Very specific output specifications
 - Needed to line up with the input character to character
 - Exact solutions
- Models did not fine-tune to the problem very well

Takeaways

- Neural Networks struggle to adapt to logical tasks
- Problems that seem more complex to humans perform the same using neural networks
- Fine-tuning struggles when a very specific output is required
- Increases in model size generally improve performance, but with diminishing returns

Future Considerations

- Using a smaller model as a base for fine-tuning
- Scale the model smarter focusing on larger feed forward networks and training
- Looking for a more homogeneous dataset and presenting cleaner data as input
- Setting aside longer periods of time to train models in depth