Dynamic Spectra Sequence Modelling with Transformers

Zach Yek[1,2], Steve Croft[1], Yuhong Chen[1]

[1]Department of Astronomy, University of California Berkeley, Berkeley CA 94720, USA

[2]Department of Physics, State University of New York at Fredonia, Fredonia NY 14063, USA

Introduction

In the past, Breakthrough Listen has leveraged mostly Computer Vision-based techniques (e.g. Convolutional Autoencoders) to generate vector embeddings used to distinguish between some RFI [1, 2]. However, most of these samples were still projected too closely in the embedding space to separate them from the main centroid [3].

Motivated by the recent success of sequence models applied to SETI searches (e.g. [4]), along with the observation that dynamic spectra are inherently sequential, we find it valuable to apply sequence modelling techniques commonly used in the field of Natural Language Processing, such as the Transformer model [5].

Expected Outcomes

While work is still ongoing, we expect to soon have a semi-supervised model for converting dynamic spectra into vector embeddings, which we can then use to apply clustering and anomaly detection methods to help us search for and detect anomalous signals present in the dynamic spectra. We expect this model to outperform techniques that Breakthrough Listen currently uses, due to the reasons outlined in previous sections.

Aims

Due to its architecture, as shown in Figure 1 [5], the Transformer has largely become the industry standard for dealing with sequence data, as they

- can capture positional and contextual relationships
- allow for parallel processing of data

The objective of this project is to train a Transformer using dynamic spectra and compare their performance to current models.

Methods

We've identified the Breakthrough Listen Kaggle dataset [6] as a suitable benchmark for our model's performance. These data are small regions of dynamic spectra, referred to as cadence snippets, generated using real observations from GBT. Most of these snippets contain only RFI, but certain snippets were injected with artificial signals using setigen [7] to emulate potential technosignature candidates. An example of one such snippet is given in Figure 2 [6].

During the training phase, we feed in the current timestep, along with all of the previous timesteps into the encoder block, but only the next timestep into the decoder block. This allows the model to learn the data representation in lower-dimensional space by extracting the most salient features of the data. Note that the following changes will be made to the standard architecture presented in Figure 1:

1. Remove input embedding layers
2. Replace SoftMax layer with a Sigmoid layer

Aside from tuning the hyperparameters to achieve an optimal loss function, we plan to experiment with different normalization schemes:

1. Batch normalization: normalizing across each sample
2. Layer normalization: normalizing across each feature

We're also planning to experiment with different input formats:

1. Concatenate all six snippets into a 30 minute observation, effectively having one big input vector
2. Leave the six 5 minute snippets as is, which yields six smaller input vectors

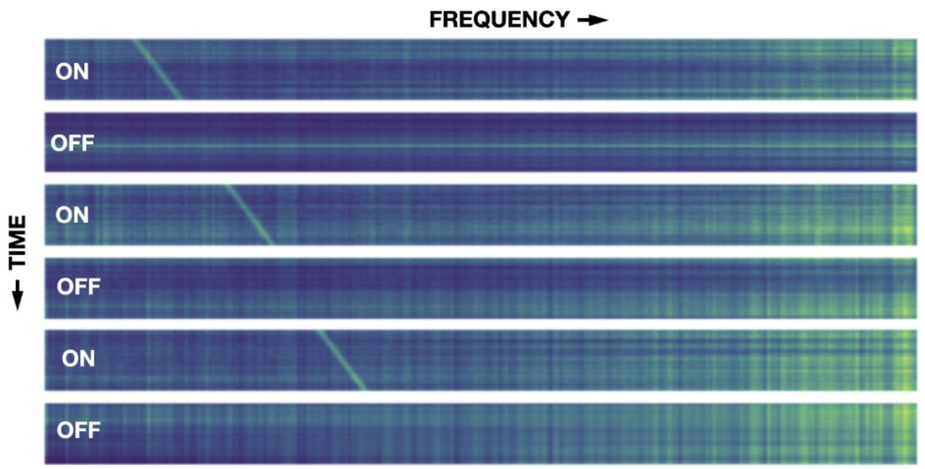

Figure 2

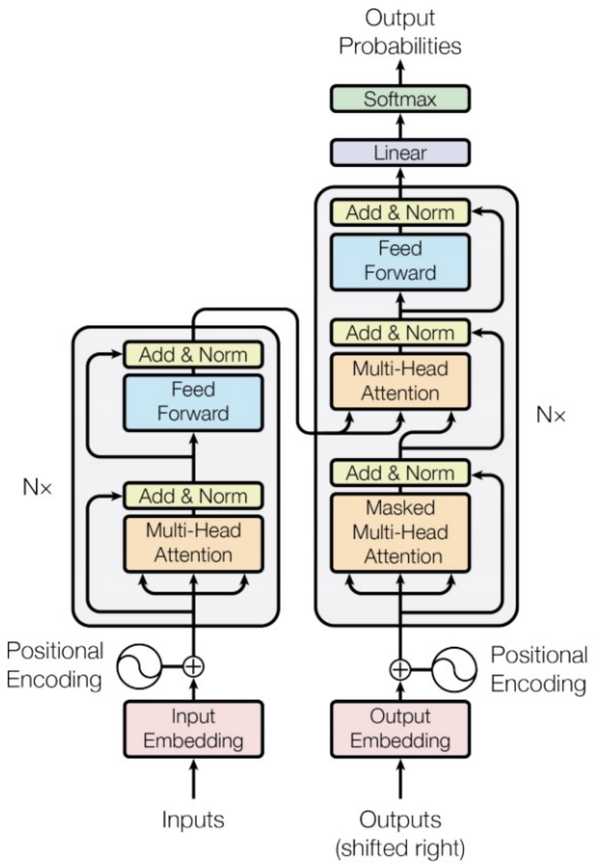

Figure 1

References

[1] Chen, Y., *et. al.* (2019). *SETI Energy Detection*. GitHub. https://github.com/FX196/SETI-Energy-Detection.

[2] Yek, Z. (2021). *MNIST Auto Classifier*. GitHub. https://github.com/zachtheyek/autoSETI/blob/master/MNIST_auto_classifier.ipynb.

[3] Chen, Y. (2020). *Deep Residual Embedded Clustering For Dynamic Spectra*. Google Slides. https://docs.google.com/presentation/d/1VuUKkjk9pY9OdylONXrpDd_OdVrxp2R3XrmiXhDxuRE/edit?usp=sharing.

[4] Zhang, Y., *et. al.* (2019). *Self-Supervised Anomaly Detection For Narrowband SETI*. arXiv. https://arxiv.org/abs/1901.04636.

[5] Vaswani, A., *et. al.* (2017). *Attention Is All You Need*. arXiv. https://arxiv.org/abs/1706.03762.

[6] *SETI Breakthrough Listen - E.T. Signal Search*. (2021). Kaggle. https://www.kaggle.com/c/seti-breakthrough-listen.

[7] Brzycki, B., *et. al.* (2018). *Setigen*. GitHub. https://github.com/bbrzycki/setigen.

Acknowledgments

This work was made possible with financial support from the Breakthrough Prize Foundation, which funds the Breakthrough Initiatives, which manages Breakthrough Listen. We thank the Kaggle staff for their support with hosting the Breakthrough Listen Signal Search competition. The technical supervision provided by Steve Croft and Yuhong Chen was also greatly appreciated.