1. Design

The chosen design refinement was to implement a 1-dimensional convolutional autoencoder which processes the *lightcurve* flux values. As depicted in Figure 1, the layer structure was similar to the example implementation, however instead of processing the raw flux from each image, the autoencoder input was the total flux value after the aperture mask was applied. The purpose of the autoencoder was to filter high frequency noise from the 1D *lightcurve* flux values and increase the signal-to-noise ratio (SNR). Training and testing was completed with the same dataset, however layer 1 (Gaussian noise) was only active during training. This Gaussian noise layer ensured that the autoencoder did not overfit. The standard deviation of the Guassian noise was optimised via an iterative search to produce a autoencoder with the highest *signal-to-noise ratio* (SNR) after the data was fitted with *transit-least-squares*.

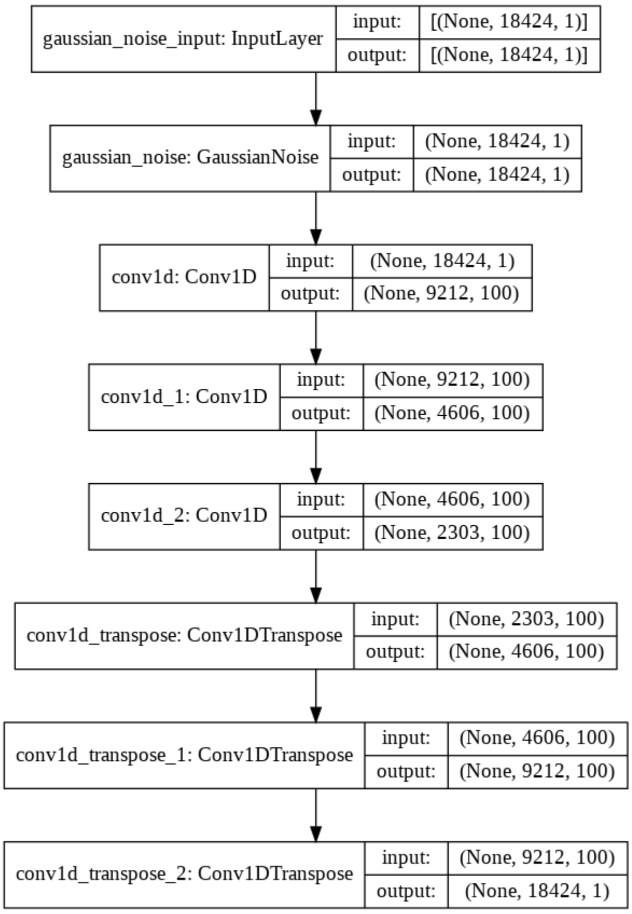

Figure 1: 1-Dimensional Convolutional Autoencoder Structure

The hypothesis was that the convolutional autoencoder would be able to 'learn' to filter the high frequency noise present in the 1D *lightcurve* flux values. This was dependent on proper training with scaled input values and a sufficient standard deviation for the Guassian noise layer. It was expected that the autoencoder would provide the best filtering when the standard deviation of the Guassian noise layer was equal to the standard deviation of the *lightcurve* flux values. Based on this theory, the optimal standard deviation for the Guassian noise layer was estimated to be 0.00075.

2. Results

As shown in Figure 2, the Autoencoder was able to increase the SNR for the processed *lightcurve*. The original baseline (control) had a SNR of 39.8dB and the autoencoder was able to increase this to 40-133dB depending on the standard deviation of the Guassian noise layer. It was found that increasing the standard deviation of the Guassian also increased the SNR. Standard deviations above 0.3 were excluded because the autoencoder began to blur important features and in most cases, no transit was detected. Figure 3 depicts the filtering capabilities of the autoencoder with an example standard deviation of 0.1 for the noise layer.

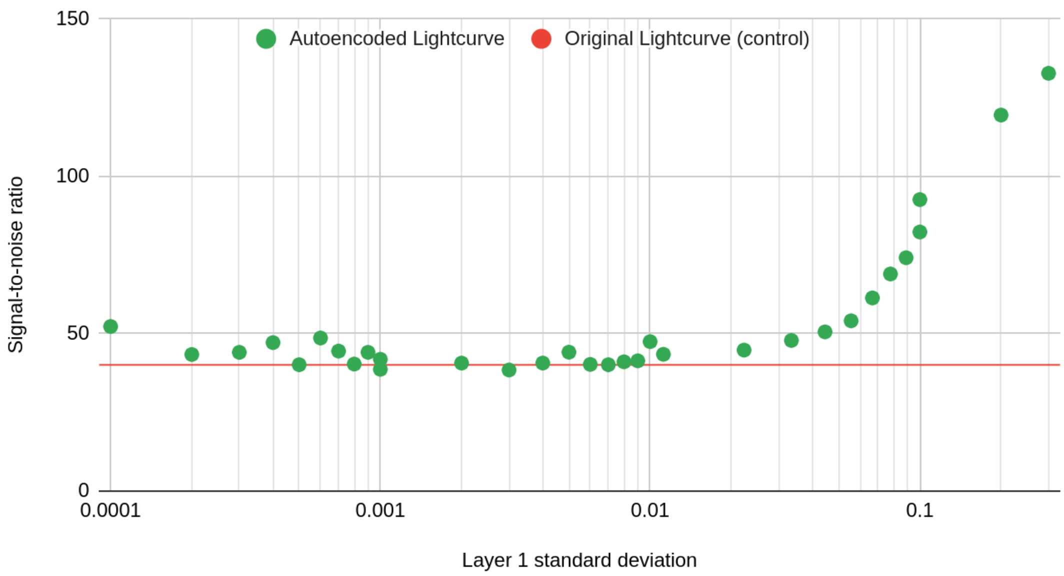

Figure 2: Iterative search for finding the optimal standard deviation for layer 1

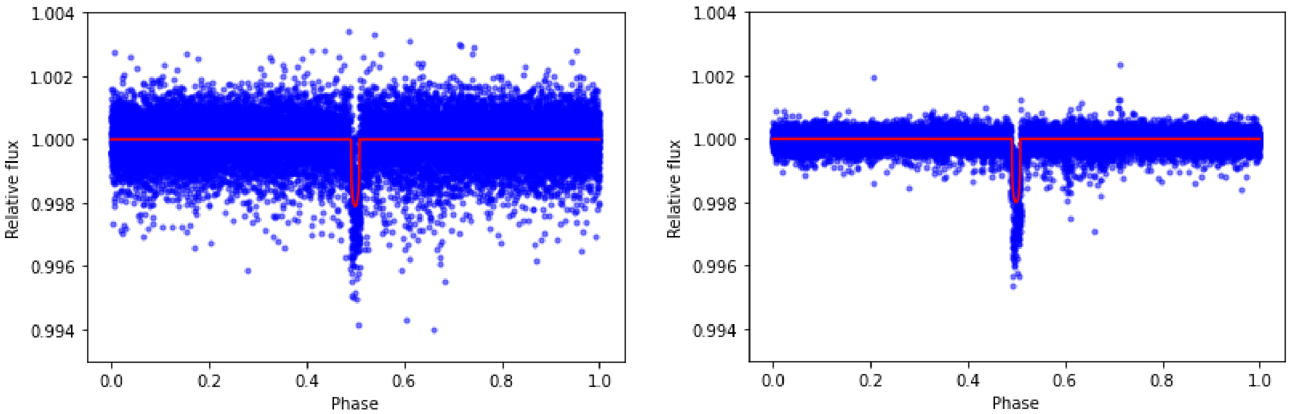

Figure 3: Comparison of the original (left) and autoencoder-processed (right) lightcurves

A positive relationship between the standard deviation of the noise for layer 1 and the SNR was observed. This trend is attributed to the autoencoder becoming increasingly better at rejecting noise in the dataset as standard deviation of the “synthetic” noise (in layer 1) increases. I.e when the autoencoder trains with more “synthetic noise”, it becomes better at rejecting the “real noise” preset in the dataset. As the autoencoder gets better at rejecting noise the SNR ratio increases as a decreasing amount of the signal is noise.

3. Conclusion

As specified in the hypothesis, the proposed convolutional autoencoder was able to “learn” to filter the high frequency noise in the *lightcurve* flux values. This was shown by a significant increase in the SNR of the processed transits. The higher SNR indicated that less noise was present in the processed transit, which means the autoencoder filter is working effectively. The original estimate of an optimal standard deviation of 0.0075 was inaccurate. This was likely because training with a larger amount of noise than what is present in the physical dataset created a more robust model.

Although this was a specific study as only SNR and layer 1 standard deviation were considered, it still contains useful learnings for those working on similar projects. The results show that varying the amount of noise in training layers allows an autoencoder-based filter to perform better and reject more noise. It is likely that the results in this study were dependent on other factors (such as the specific dataset and training parameters) however the learnings are still valid in a broad context. Specifically testing different parts of a neural network and tuning one parameter at a time is an efficient design strategy.

Future directives of this project would broaden the scope of the study. This could involve optimising the artificial noise layer present when training. Different noise models (such as dropout) could be trialled. A custom noise function to account for the non-linear noise from reflections would likely be the best solution but would require the greatest amount of time to refine. A broader approach could be to improve the overall network architecture. Parameters such as number of layers, latent layer size and stride length could be optimised to give a robust and effective autoencoder.

Appendix A: Raw Results

Layer 1 Standard Deviation	Autoencoded Lightcurve SNR (dB)
0.0001	52.20907028
0.0002	43.30360994
0.0003	43.99418572
0.0004	47.08649802
0.0005	40.06057384
0.0006	48.54929578
0.0007	44.38162975
0.0008	40.26917304
0.0009	43.99193255
0.001	41.77912408
0.001	38.59490222
0.002	40.57290211
0.003	38.37958681
0.004	40.58336025
0.005	44.04227488
0.006	40.15651534
0.007	40.05649167
0.008	40.97552791
0.009	41.29221511
0.01	47.39566854
0.0112	43.36722607
0.0223	44.70419592
0.0334	47.77329007
0.0445	50.49081329
0.0556	54.01262607
0.0667	61.2849294
0.0778	68.89840461
0.0889	74.06618882
0.1	82.2448905
0.1	92.54526646
0.2	119.4200835
0.3	132.6933117
>0.4	Nan (transit not detected)