Neural network simulating the Schuster periodogram

V.V. Topinskiy and R.V. Baluev
st076660@student.ru
r.baluev@spbu.ru

Saint Petersburg State University;
Special Astrophysical Observatory, Russian Academy of Sciences

August 19-23, 2024

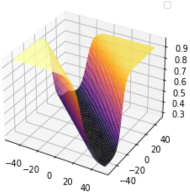
Abstract

The search for periodic components in a time series is an important aspect of data analysis. In most cases, Schuster's periodograms or Lomb-Scargle periodograms are used depending on the homogeneity of the distribution of the original data over time. Calculating spectra is not a computationally intensive task; however, difficulties arise when processing large quantities of time series data and assessing the existence of periodic components within them. For preliminary analysis of large data sets, a convolutional neural network simulating the operation of Schuster's periodogram is suitable.

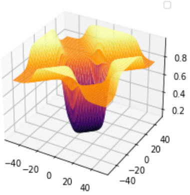
This work represents our initial step towards accelerating the primary processing of signals in the task of exoplanet detection.
We are going to use an improved version of the algorithm in the problem described above.

First try

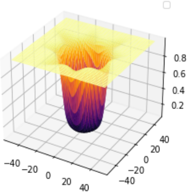
Initially, we considered single-layer neural networks that processed one pair of Fourier transform components. Later, they served as the first layer.



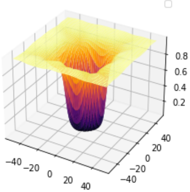
(a) 2 neurons



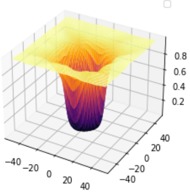
(b) 4 neurons



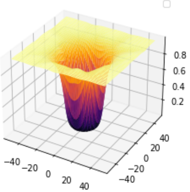
(c) 8 neurons



(d) 16 neurons



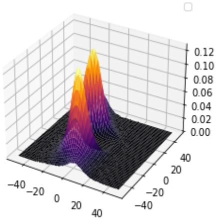
(e) 32 neurons



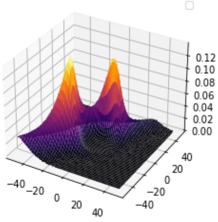
(f) 64 neurons

First try

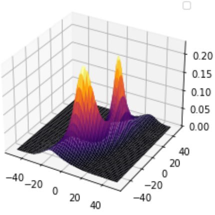
The network response that you could see on the previous slide was formed from responses of individual neurons of this type:



(a)



(b)



(c)

Problem

Problem:

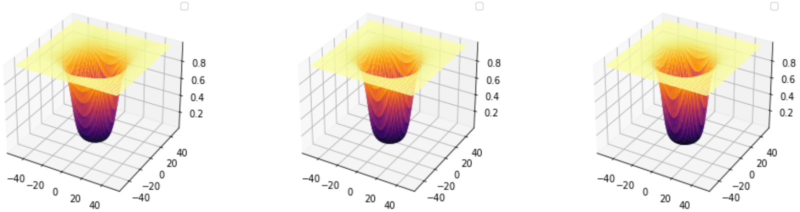
- The "ribbing" in the block response decreases with increasing number of neurons, but does not disappear.

Possible solutions:

- Further increase in the number of neurons
- Filling the weights before training in a special way

Solution

Based on a previously trained block of 128 neurons, the weights of the new block were uniformly distributed. The blocks have been transformed and now look like this.



(a) 16 neurons (b) 32 neurons (c) 64 neurons

Figure: Response graph of a neuron blocks to a pair of real and imaginary parts of a signal sample

The ribbing has noticeably smoothed out or disappeared completely.

Neural network

A two-layer perceptron was designed to determine the existence of a sinusoidal component in a signal consisting of 128 samples by means of its Fourier transform. Each layer is defined by the following formula:

$$x_k^{i+1} = f\left(\left(\sum_{j=1}^{n} w_j^i x_j^i\right) + b_k^i\right), \tag{1}$$

where f – layer's activation function (sigmoid were used), w_j^i – weight, b_k^i – bias.

Training time series were generated for network training and subsequent testing. Bayes factor, as described in the paper [1], was used to assess prediction accuracy.

A million ($N_c = 1000000$) triplets of signal parameters (Θ) are generated for each time series with noise. The following quantities are considered:

$$E_{M_2} = \sum_{i=1}^{N_c} \frac{e^{-\sum_{k=1}^{len(x)}((x[k]-f_{sin}(T[k],\Theta[i]))^2)/2}}{((2\pi)^{len(x)/2}))},$$

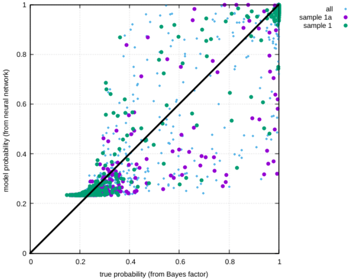
where f_{sin} – the function that generates a sinusoidal value with parameters $\Theta[i]$ and time $T[k]$.

$$E_{M_1} = \frac{e^{-\sum_{k=1}^{len(x)}((x[k]^2/2}}{((2\pi)^{len(x)/2}))}$$

Bayes factor (R) and probability of signal identification (P):

$$R = E_{M_2}/E_{M_1};\ P = R(1-R)$$

Results and mistakes



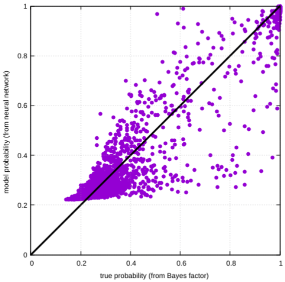
The ratio of the model probability of the existence of a signal with the theoretically possible

- The accuracy reached only 90% for series with high amplitudes
- Increasing accuracy required increasing the number of neurons, which in turn resulted in numerous "extra" connections in the network that needed to be optimized through training, consequently increasing the amount of required data
- It was decided to restructure the network to reduce the number of neural connections while improving accuracy. In this case, the best alternative was to introduce convolutional layers into the structure.

Convolutional neuron network

Replacing regular layers with convolutional layers reduces the number of trainable connections and allows for an increase in the number of neurons in each layer. To optimize training, weights from pre-trained models' layers were used. The network structure now consists of 2 convolutional layers processing a pair of Fourier components (Figure 3 (a).), a technical flattening layer, and 2 regular layers responsible for finding the maximum and outputting the result as the probability of a sinusoidal component in the signal. The neural network was implemented using Python 3.8, and the network structures were taken from the module keras. Training took place over 300 epochs, with the adadelta optimizer, accuracy metric, and binary cross-entropy loss function. The choice of optimizer was based on its precise and rapid weight minimization, as determined through empirical testing.

Tests and results



The ratio of the model probability of the existence of a signal with the theoretically possible

- Tests were conducted on synthesized datasets with approximately $N \sim 10^6 - 10^7$.
- The amount of data containing a sinusoidal component and data consisting solely of noise was equal. This volume allowed achieving a signal detection accuracy of 99% of the theoretical maximum with a small number of trainable network neurons.
- Additionally, no overfitting issue affecting the network's response was observed.

Conclusion

This work presented a brief description of a convolutional neural network model solving the signal detection problem. The described structure is currently not well-suited for real data; hence work is underway to expand its functionality, specifically introducing weights to time series and processing non-uniform series (simulating the operation of Lomb-Scargle periodograms). The synthesis of training datasets will also be revised for more efficient training. These steps will enable obtaining results from real data.

References

Roman V. Baluev, *Comparing the frequentist and Bayesian periodic signal detection: rates of statistical mistakes and sensitivity to priors*. preprint (2022), available at https://arxiv.org/abs/2203.08476.