BAAA, Vol. 64, 2023
R.D. Rohrmann, C.H. Mandrini, C.E. Boeris & M.A. Sgró, eds.

Asociación Argentina de Astronomía
Boletín de artículos científicos

Wiener Filter for cosmic microwave background maps using neural networks

M.B. Costanza[1,2], C.G. Scóccola[1,2] & M. Zaldarriaga[3]

[1] *Facultad de Ciencias Astronómicas y Geofísicas, UNLP, Argentina*
[2] *Consejo Nacional de Investigaciones Científicas y Técnicas, Argentina*
[3] *School of Natural Sciences, Institute for Advanced Study, EE.UU.*

Contact / belen@fcaglp.unlp.edu.ar

Resumen / Estudiamos una red neuronal convolucional llamada WienerNet la cual aplica el filtro de Wiener a mapas del Fondo Cósmico de Radiación (FCR) con el objetivo de reducir el ruido presente en dichos mapas. Presentamos el funcionamiento de la red neuronal, y comparamos los resultados con los obtenidos al aplicar el filtro de Wiener con el método tradicional, que utiliza el gradiente conjugado. A su vez, mostramos la eficiencia de la aplicación de WienerNet respecto del método tradicional, el cual constituye un cuello de botella en el análisis de datos del FCR. Para este propósito, aplicamos la red neuronal a mapas del FCR con diferentes número de píxeles y diferentes modelos de ruido, y comparamos la eficiencia computacional en cada caso.

Abstract / We studied a convolutional neural network called WienerNet which applies the Wiener Filter to Cosmic Microwave Background (CMB) maps, whose objective is to reduce the noise present in those maps. We present how the neural network works, and compare its results to those obtained when applying the wiener filter with the traditional method, which uses the conjugate gradient. Also, we show the efficiency of WienerNet with respect to the traditional method which constitutes a bottleneck in the data analysis of the CMB. For these purposes, we applied the neural network to CMB maps with different numbers of pixels and different noise models, and we compared the computational efficiency in each case.

Keywords / cosmic background radiation — cosmological parameters — early universe — methods: numerical — methods: statistical

1. Introduction

After the Big Bang, when the temperature of the Universe drops to $\lesssim 10^4 K$ due to the expansion, the electrons and protons combined to form neutral hydrogen during the recombination epoch, at redshift $z \sim 1100$ (Dodelson, 2003). The photons decoupled from the plasma, traveled freely through space and constitute the Cosmic Microwave Background (CMB). Those photons have a blackbody spectrum with temperature $T = 2.725K$ and presents anisotropies of the order of 10^{-5} due to small perturbations in the primordial plasma. The statistical properties of the temperature distribution are described by the angular power spectrum which is a function of the multipole moment l. The study of the angular power spectrum allows us to determine the cosmological parameters of the ΛCDM model. For that purpose, it is necessary to conduct statistically optimal data analysis.

The Wiener Filter (WF) is an optimal filter to reduce the noise present in simulated or real CMB maps and reconstruct the original signal underneath. Nevertheless, the standard WF method, which uses the conjugate gradient (CG), is a bottleneck in data analysis due to its large computational cost. An implementation of the WF, the NIFTY code, is given in Selig et al. (2013).

In this article, we study the implementation of the WF with a neural network called WienerNet with different noise models and CMB maps with different number of pixels, and study the efficiency of this neural network with respect to the standard method (i.e., with NIFTY).

2. Wiener Filter

The WF formalism is widely used in Cosmology for reconstructing the underlying signal, either for the estimation of the matter power spectrum or the angular power spectrum of the CMB. In this article, we are interested in the reconstruction of the temperature distribution given a noisy dataset of temperature CMB maps.

The measurements d of an underlying field s that we want to estimate is a linear combination of the field, where R is the response matrix of the measurement procedure and ϵ is the data uncertainty:

$$\mathbf{d} = \mathbf{R}\mathbf{s} + \epsilon \tag{1}$$

The reconstruction of the underlying field is a linear combination of the measurements

$$s^{MV} = \mathbf{F}\mathbf{d}, \tag{2}$$

where the matrix F is the WF matrix obtained by minimizing the variance of the residual:

$$\mathbf{F} = \mathbf{S}(\mathbf{S} + \mathbf{N}_\sigma)^{-1}\mathbf{R}^{-1}. \tag{3}$$

In the particular case where the underlying signal is a gaussian random field, the WF estimator obtained by minimizing the variance of the residuals coincides with the Bayesian estimator that maximizes the conditional

Oral contribution

probability of the signal given the data (Zaroubi et al., 1995):

$$P(\mathbf{s}|\mathbf{d}) \propto \exp\left[-\frac{1}{2}(\mathbf{s}^\dagger\mathbf{S}^{-1}\mathbf{s} + (\mathbf{d}-\mathbf{Rs})^\dagger\mathbf{N}^{-1}(\mathbf{d}-\mathbf{Rs}))\right]. \quad (4)$$

The WF estimator is, therefore, the optimal reconstruction of the signal because it is equal to the most probable configuration of the field given the data.

3. Machine Learning: Neural networks

In supervised machine learning algorithms, the model learns how to combine the features to develop useful predictions on new data. A machine learning system (ML) is fed with labeled examples composed of features and targets (the true values that the models pretend to predict). In that sense, during the training of the model, the meaning of learning is to find the weights that allow the model to relate correctly the features with the labels by minimizing a cost function.

Neural networks are a specific type of ML models, called "deep learning" (Chollet, 2017), that can be built with several layers and nodes (neurons) with non-linear functions depending of the problem. There are many types of neural networks, WieneNet* is an autoencoder neural network composed of two dimensional convolutional layers, which attempts to simulate the WF (Münchmeyer & Smith, 2019). It receives noisy images of the CMB as inputs and returns a reconstructed map of the original signal.

4. Results

We analyzed the results of WienerNet for CMB maps with different numbers of pixels starting with a simple case of 28×28 pixels and 30 arcmin of angular resolution (i.e., the size of the map is $14^\circ \times 14^\circ$). To simulate the temperature maps, we use the libraries CAMB (to generate the power spectrum) and HEALPY (to simulate the map given the power spectrum). We then use a flat sky approximation and homogeneous noise. We adapted the neural network code to the specific size of the map by changing the number of encoders and decoders.

We train the neural network with a training set of 4000 maps and a validation set of 1000 maps. We then evaluate the trained WF model with a test set of 300 maps. The WienerNet code was implemented in TENSORFLOW 2.4 and KERAS.

We applied the neural network to the test set obtaining 300 filtered maps with WienerNet. We also calculated the exact result of the WF with the conjugate gradient method to each map of the test set, and then we computed the difference pixel by pixel between the exact result of WF and the predictions given by the neural network in order to evaluate how much both methods differ. The distribution of these residuals, for a typical map, has a mean of -3.94×10^{-8} and a standard deviation equal to 25.07. Computing the same for the whole test set, the distribution of the means has a mean value of 2.98×10^{-09}, with a dispersion of 1.4. More details about the performance of WienerNet with respect to the exact WF are found in (Münchmeyer & Smith, 2019).

*https://github.com/moritzmunchmeyer/wienernet

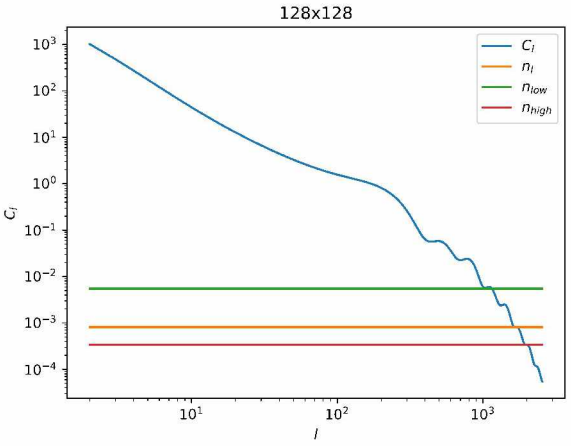

Figure 1: Power spectrum of the CMB for 128×128 maps and three white noise models.

We studied the efficiency of making the WF on the test set with the neural network with respect to the CG method and the scaling with the number of pixels and noise models. We simulated maps with several number of pixels: 56×56, 128×128, 256×256 and 512×512, with angular size for the map of $10^\circ\times 10^\circ$.

We also simulated homogeneous noise models with three different noise levels. We characterize the noise level by the angular scale $\hat{\ell}$ at which the power spectrum of the white noise cuts the power spectrum of the signal. We consider different values for $\hat{\ell}$, in order to have more, the same, or fewer number of modes with high signal-to-noise ratio, as can be seen in figure 1. The value of $\hat{\ell}$ is different for different map sizes.

Table 1 and figure 2 shows the computational time required to compute the WF with the neural network, in seconds, calculated with CPU. For 56×56, it takes 1 seconds, and for 512×512, it takes less than a minute.

Table 1: Time CNN [sec]

l	56	128	256	512
$\hat{l}$	1.06	7.01	41.9	54.9
$\hat{l}_{low}$	1.11	5.71	43.11	58.49
$\hat{l}_{high}$	1.07	6.51	41.93	59.52

On the other hand, Table 2 and and figure 3 shows the computational time for computing the WF with the traditional method CG, in seconds, also with CPU. In the case of 56×56 it takes 1.5 seconds, but rapidly increases, and from 128×128 on, it significantly exceeds the computational time required by the neural network for calculating the WF. It takes 2 hours for 256×256 and some days for 512×512. It can be seen from figure 3 that it takes more time to compute the WF with CG in the case of lower noise level corresponding to the

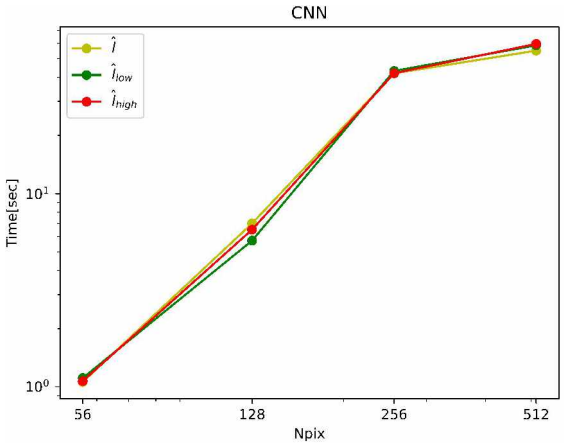

Figure 2: Computational time required to compute the WF with the neural network scaled with the number of pixels.

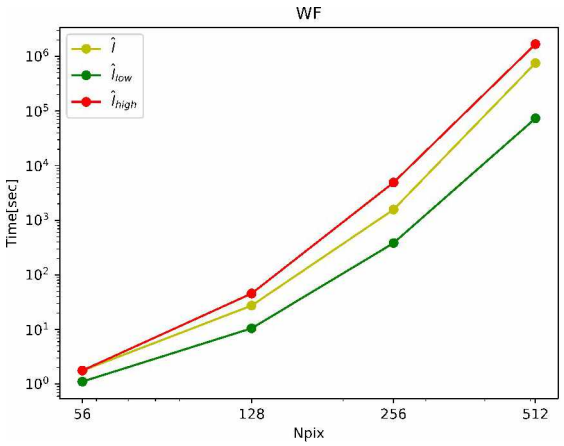

Figure 3: Computational time required to compute the WF with the CG scaled with the number of pixels.

scale $\hat{l}_{high}$ because the signal spectrum is predominant over noise as can be seen in figure 1, making the exact WF slower. This effect is not appreciated in the case of performing the WF with the neural network as can be seen in figure 2 where the computing time is almost the same in the three noise levels.

Table 2: Time CG [sec]

l	56	128	256	512
$\hat{l}$	1.76	27.4	1570	759600
$\hat{l}_{low}$	1.11	10.5	384	73500
$\hat{l}_{high}$	1.77	45.6	4909	1682100

Finally, we studied the efficiency of the neural network in the case of an inhomogeneous noise model. To this end, we simulate maps of 512×512 number of pixels with angular size $20° \times 20°$ and use a realistic noise model taken from the Planck maps Planck Collaboration et al. (2020). In particular, we use the variance map from the 143 GHz frequency channel, that we extract from the Planck Legacy Archive** as it can be seen in figure 4. In this case, it was necessary to change the original architecture of the neural network, to include the noise variance map. For inhomogeneous noise, the improvement in the time performance is very large. Indeed, the time for computing the WF with the neural network is equal to 81.77 sec and the time for computing the WF with the CG method is 13617.88 sec.

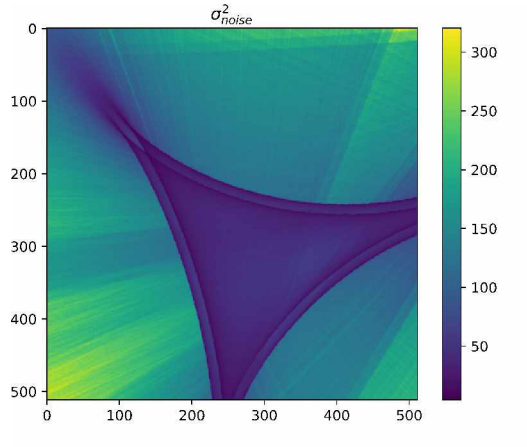

Figure 4: Map variance extracted from Planck. The axes indicate the pixel number, while the value of the variance for each pixel is color-coded as indicated in the color bar.

5. Conclusions

In this work, we have shown that the WienerNet neural network is able to successfully compute the WF of CMB temperature maps and the results are in agreement with the results of computing the WF with the exact method. The computational cost of making the WF with the CG method largely exceeds the computational cost of the neural network, showing that the neural network is more efficient to perform the WF with different noise properties. This neural network can be used in the pipeline of the CMB data analysis.

Acknowledgements: CGS and MBC acknowledge funding from CONICET (PIP-2876), and Universidad Nacional de La Plata (G11-157 and G11-175), Argentina.

References

Chollet F., 2017, *Deep Learning with Python*, Manning Publications, New York, NY
Dodelson S., 2003, *Modern Cosmology*, Academic Press, Elsevier Science
Münchmeyer M., Smith K.M., 2019, arXiv e-prints, arXiv:1905.05846
Planck Collaboration, et al., 2020, A&A, 641, A3
Selig M., et al., 2013, aap, 554, A26
Zaroubi S., et al., 1995, ApJ, 449, 446

**http://pla.esac.esa.int/pla/#maps