Astronomy and Computing 18 (2017) 35–39

Contents lists available at ScienceDirect

Astronomy and Computing

journal homepage: www.elsevier.com/locate/ascom

Full length article

Radio frequency interference mitigation using deep convolutional neural networks[☆]

J. Akeret[a,*], C. Chang[a], A. Lucchi[b], A. Refregier[a]

[a] ETH Zurich, Institute for Astronomy, Department of Physics, Wolfgang Pauli Strasse 27, 8093 Zurich, Switzerland
[b] ETH Zurich, Data Analytics Lab, Department of Computer Science, Universitaetstrasse 6, 8092 Zurich, Switzerland

ARTICLE INFO

Article history:
Received 29 September 2016
Accepted 12 January 2017
Available online 21 January 2017

Keywords:
Radio frequency interference
RFI mitigation
Deep learning
Convolutional neural network

ABSTRACT

We propose a novel approach for mitigating radio frequency interference (RFI) signals in radio data using the latest advances in deep learning. We employ a special type of Convolutional Neural Network, the `U-Net`, that enables the classification of clean signal and RFI signatures in 2D time-ordered data acquired from a radio telescope. We train and assess the performance of this network using the `HIDE` & `SEEK` radio data simulation and processing packages, as well as early Science Verification data acquired with the 7m single-dish telescope at the Bleien Observatory. We find that our `U-Net` implementation is showing competitive accuracy to classical RFI mitigation algorithms such as `SEEK`'s SumThreshold implementation. We publish our `U-Net` software package on GitHub under GPLv3 license.

1. Introduction

The radio band is becoming one of the most promising wavelength windows for cosmology. In particular, observations of the 21 cm neutral hydrogen line allow us to probe the high-redshift Universe, which is not easily accessible with other wavelengths (Pritchard and Loeb, 2012). In addition, radio band data provides important information for foreground studies of cosmic microwave background, and also Galactic astronomy (Chang et al., 2017). Ongoing and future experiments such as LOFAR (Haarlem et al., 2013), GMRT (Paciga et al., 2013), PAPER (Ali et al., 2015), CHIME (Bandura et al., 2014), BINGO (Battye et al., 2013; Battye et al., 2012), HERA (Pober et al., 2014), Tianlai (Chen, 2012), and the SKA (Mellema et al., 2015) aim to carry out wide-field surveys in the radio band that covers large portions of the sky.

One of the main challenges in all these surveys is the radio frequency interference (RFI) contamination to the data (Offringa et al., 2010a). RFI can originate from a wide variety of human produced sources such as satellites (GPS, geostationary, TV, etc.), cell phones, and air traffic communication. Different sources of RFI display different frequency and time-dependencies, causing the overall RFI signal to be complex and difficult to model Fridman and Baan (2001). If the RFI signal is strong and mixed with the astronomical signal of interest, the data cannot be used and will need to be masked.

To minimize the RFI contamination to data, radio telescopes are normally built in remote locations that are protected against major human-made emission sources. Some level of hardware improvement such as ground-shielding and band-pass filters can also reduce the input of RFI. However, in almost all situations, RFI masking in the analysis software will still be needed.

The goal of any RFI masking algorithm is to minimize the amount of data lost while ensuring low RFI contamination. This procedure typically relies on the common assumption that the morphological characteristics of RFI in the 2D plane of time and frequency (the raw data format of standard spectrometers) are different from that of astronomical signals. Astronomical signals are usually broad-band and vary smoothly over long time-scales, while RFI appears as high-intensity pixels localized in the time–frequency plane or is sometimes also periodic in time. Existing RFI mitigation algorithms typically fall into three categories. The first category attempts to identify the characteristics of RFI through linear methods such as Singular Vector Decomposition (SVD) (Offringa et al., 2010a) or Principle Component Analysis (PCA) (Zhao et al., 2013). These methods work well if the RFI pattern exhibits a repeated pattern over time and frequency, but cannot handle with more stochastic signals such as the ones caused by irregular satellites. The second category uses threshold-based algorithms such as cumsum (Baan et al., 2004) and SumThreshold (Offringa et al., 2010a), where the RFI is defined as pixels above some threshold in the smoothed 2D time–frequency plane. Despite their simplicity, these methods are fairly reliable and can be quite effective. In particular, SumThreshold is the most widely used algorithm in existing radio data processing pipelines (Offringa et al., 2010a; Offringa et al., 2010b; Peck and Fenech, 2013; Akeret et

☆ This code is registered at the ASCL with the code entry ascl:1611.002.

* Corresponding author.
E-mail addresses: joel.akeret@phys.ethz.ch (J. Akeret), chihway.chang@phys.ethz.ch (C. Chang), aurelien.lucchi@inf.ethz.ch (A. Lucchi).

http://dx.doi.org/10.1016/j.ascom.2017.01.002
2213-1337/

al., 2017). The third category uses traditional supervised machine-learning techniques such as K-nearest neighbor and Gaussian mixture models to cluster RFI signals (Wolfaardt, 2016). For these methods to achieve a sufficient classification accuracy, a careful feature selection process has to be performed prior to the application. While these three classes of methods have encountered a significant success in astronomy, somewhat more advanced techniques in machine learning have not been explored.

One approach that has shown promising results in the area of machine learning are deep neural networks. In the recent years, they outperformed state-of-the-art techniques in various classification tasks such as biomedical image segmentation (Ronneberger et al., 2015) or natural language processing (Collobert et al., 2011). Although the concept of artificial neural network has been around for many years, their current preeminence can be mostly attributed to recent advances in customized hardware (especially GPUs) as well as the development of open source deep learning software packages.[1]

A particular successful type of network is the convolutional neural network (CNN) (e.g. Krizhevsky et al., 2012; Collobert and Weston, 2008). Typically, CNNs have been used to detect objects in images (without having any exact prior knowledge of where the object appears in the image). These networks have also recently been extended to the problem of image segmentation, for which a class label is assigned to each pixel in an input image. One example of this segmentation network is the `U-Net` Ronneberger et al. (2015). In this paper we apply this type of CNN to identify and mitigate RFI in time-ordered-data (TOD) of a single-dish radio telescope. To the best of our knowledge, this is the first application of deep learning techniques to this class of problems.

This paper is organized as follows. In Section 2 we describe the basic architecture and design of the `U-Net`. In Section 3, we apply the CNN to mitigate RFI on data taken at the Bleien Observatory. This includes a discussion of the performance of the CNN both on simulated and observed data. We then conclude in Section 4. Information for downloading and installing our implementation of the `U-Net` package is described in Appendix A. In Appendix B we explain how to use the package.

2. Proposed approach

2.1. Network architecture

The `U-Net` Ronneberger et al. (2015) extends the architecture of conventional CNN's. Typically, CNN's extract image features by repeatedly applying convolutions on the input image followed by an activation function and a downsampling operation. These nested operations let the network build a conceptual hierarchy of the content present in the training images. Some similarities can be drawn to the human visual system where the early layers extract small, localized features such as edges while deeper layer combines these extracted edges into more complex representations. Note that the downsampling operations present in a CNN lead to a contraction of the information flowing through the network. This makes conventional CNNs not well suited for image segmentation.

Instead of relying on a traditional architecture, the `U-Net` extends the contracting path of a CNN by a symmetric expansive path. As shown in Fig. 1, the information on the extracted complex features (orange box) from the pooling path are propagated to the higher layers by several upsampling operations. The downsampling path followed by the upsampling path resembles a U-shape leading to the name of this network architecture.

We have reimplemented the original `U-Net` Ronneberger et al. (2015), written in Caffe, with the open source library `Tensorflow` following its exact architecture. Our `Tensorflow` `U-Net` implementation is written in Python with maximal flexibility in mind. The package is published on GitHub[2] under GPLv3 license and can be used for various classification tasks (see Appendix A and Appendix B for installation instructions and usage examples). In the contracting path we apply in each layer two consecutive unpadded convolutions both followed by a rectifier linear unit (ReLU) activation and a 2×2 max pooling downsampling operation. At each layer we double the number of extracted features. In the expansive path we replace the max pooling by an up-convolution that halves the number of features from the previous layer and concatenate the result with the features from the corresponding contraction layer. Finally, we apply a 1×1 convolution to map the features from the last layer to the number of class labels i.e. to a binary decision if a pixel is contaminated or not. To obtain the probability of a pixel to belong to a certain class we convert the resulting output map with a pixel-wise soft-max layer. The RFI mitigation is done by inputting the TOD and applying a threshold on the predicted probability of each pixel to be contaminated with RFI.

2.2. Training the network

We train the parameters of the `U-Net` using the early Science Verification data acquired at the Bleien Observatory Chang et al., 2017. This data set was collected using a 7m single-dish telescope operating in drift-scan mode with a frequency range of 990– 1260 MHz. We have processed the data with the `HIDE` & `SEEK` radio data processing pipelines described in Akeret et al. (2017). The pipeline employs the SumThreshold algorithm to mask pixels contaminated with RFI. SumThreshold is a widely used iterative algorithm that is gradually building a mask to flag the unwanted signal. It follows the underlying assumption that the astronomical signal is relatively smooth, both, in time and frequency direction. While RFI signal exhibits patterns with sharp edges, the algorithm gradually improves a model of the astronomical signal and masks values lying above a certain threshold after subtracting this model from the data. It starts with localized, strong RFI bursts and extends the mask by gradually analyzing the neighboring pixels (Akeret et al., 2017). The parameters we adopt for the SumThreshold algorithm here are based on the procedure developed in Akeret et al. (2017). We use the SumThreshold mask as ground truth to train the neural network as well as to evaluate the performance of the network on a separate test set. We note, however, that the RFI mask produced by SumThreshold is not perfect. It has a high false-positive-rate i.e. many pixels are incorrectly flagged as RFI. Some RFI detection pipelines have refined this technique, e.g. by using a scale invariant dilation operation Offringa et al. (2012). This can improve the flagging performance of the algorithm. However, we demonstrate in this paper that our `U-Net` model is robust to this noise in the ground truth and is capable of correctly distinguishing between non-contaminated and contaminated pixels.

We explore the effects of various parameters on the classification performance and processing time. Here we report the effect of the parameters that most influence the performance such as the depth of the network (i.e. the number of layers), the number of features extracted in the first layer, and the size of the convolution kernels. We optimize a cross-entropy loss function to train the network parameters using a momentum-based stochastic gradient decent with an exponentially decaying learning rate with an initial value of 0.2. We initialize the weights of the network using a truncated normal distribution following the recommendation for

[1] We here will be using `Tensorflow`, a recent deep learning framework released by Google.

[2] http://github.com/jakeret/tf_unet.