Phased Array adaptive beamforming using Deep Learning Techniques

PAF & Advanced Receiver Workshop 2022

CSIRO

INAF
ISTITUTO NAZIONALE
DI ASTROFISICA

Goals

Take advantage of using Deep Learning techniques to solve the problem of the PAF synthesis for ”near” real-time.

Investigate the possibility of using Convolution Neural Networks (CNNs) to extract features from the radiation diagrams of the patterns.

Compare this new approach with a chosen conventional optimization system.

Summary

01 Introduction

02 Pattern images generation

03 Train & test

04 Final considerations

The problem

Conventional optimization methods, for their intrinsic iterative nature, may require a long time to converge to an optimal or sub-optimal solution, because a large number of iterations or complex computations for matrix inversions is typically required.

Optimization algorithms

Phase 1

- ➢ Desired Beam pattern;
- ➢ Initial beam pattern.

Phase 2

Use a cost function to minimize difference between the desired pattern and the resulting patterns.

Phase 3

Generate the weights that are applied to the antennas to form the desired beam.

Phased Array System Toolbox

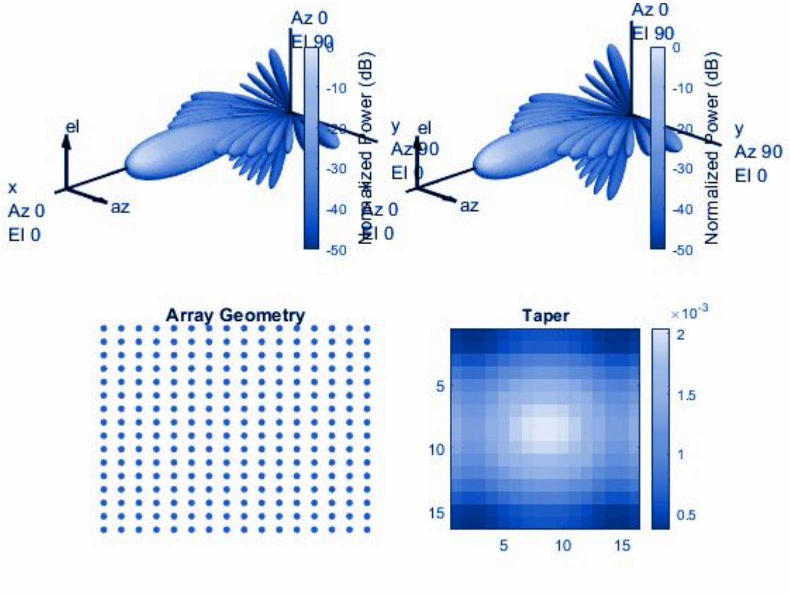
Phased Array System Toolbox provides algorithms and apps for designing and simulating beamforming systems and sensor arrays for wireless communications, radar, sonar, acoustic, and imaging applications.

https://it.mathworks.com/products/phased-array.html

Phased Array System Toolbox II

The Phased Array Toolbox brings some simplifications, the most significant are:

- The toolbox simulates the beam in the sky and not the one reflected by the radio telescope currently this feature is under development;
- The antennas are modeled as belonging to an infinite array so the mutual coupling is the same for all but in a realistic case isn't.
- The toolbox has been designed more for transmission than reception systems



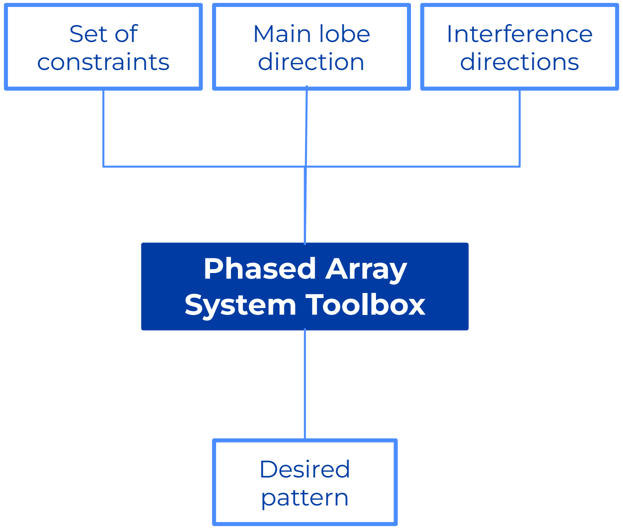
Pattern images generation

To derive the pattern, the toolbox needs:

- Main lobe direction;
- Interferences direction.

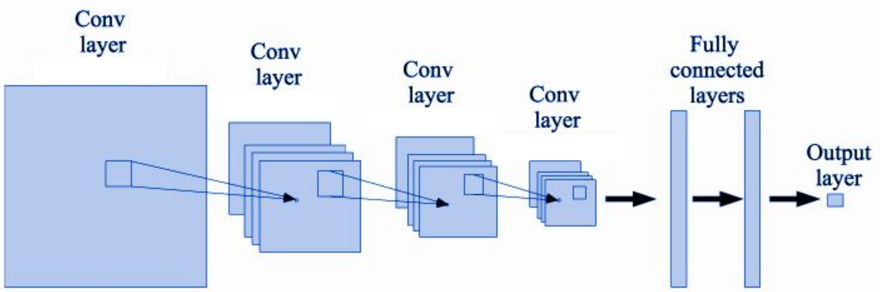
A set of desired constraints e.g:

- Maximize the directivity;
- Suppress interferences below mainlobe;
- Keep sidelobe levels below mainlobe.



Convolutional neural network

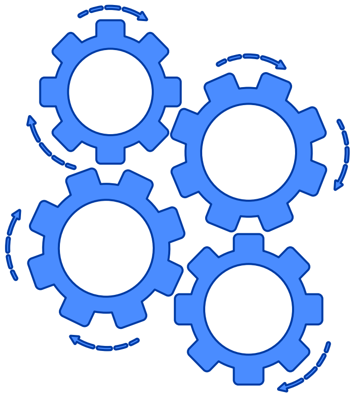
The first layers of the network are convolutional to reduce the high dimensionality of images without losing their information, while the lasts are fully connected. The final layer of the network is a fully connected layer with no activation function that outputs the weights vector.



Dataset generation I

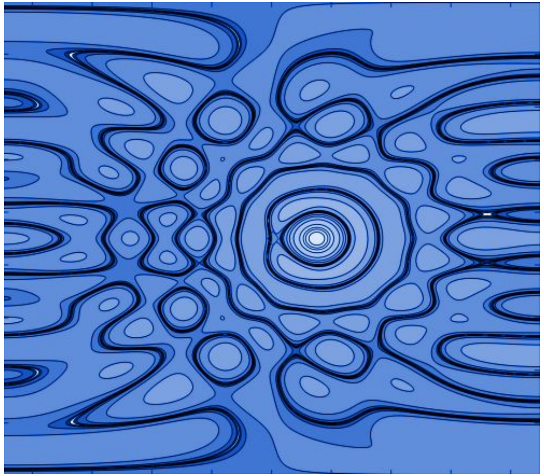
Write a script to generate patterns varying the direction for the mainlobe and the origin of the interferences.

The dataset needs to be as varied as possible so the network will be able to understand the problem, especially for mitigating the interferences.



Dataset generation II

For each pattern, will be produced the corresponding radiation diagram and the weights for the antennas to obtain it, using a classical optimization system taken as a reference. In this way will be easy to build the necessary ground truth.



$$w = \begin{bmatrix} 1 \\ e^{-\frac{j\pi\sqrt{2}}{2}} \\ e^{-\frac{j3\pi\sqrt{2}}{2}} \\ \cdots \\ e^{-j4\pi\sqrt{2}/2} \end{bmatrix}$$

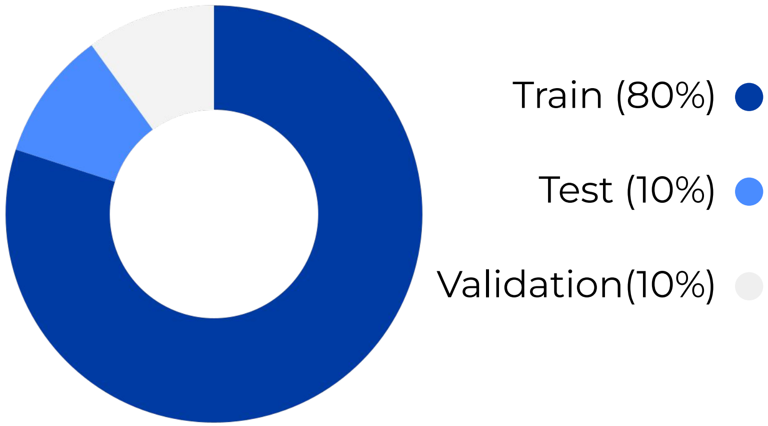
Image from: S. Bianco et al. "AESA Adaptive Beamforming Using Deep Learning," 2020 IEEE Radar Conference (RadarConf20), 2020.

Training I

The dataset obtained as described in the previous slides will be divided into 3 subgroups:

- 80% training;
- 10% validation;
- 10% test;

The validation set will be used to monitor the progress of training at the end of each epoch.



Training II

Generate random patterns with different mainlobe direction and interferences

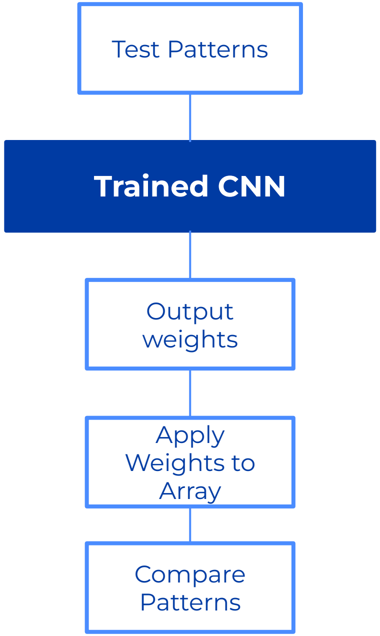
Calculate the weights for each pattern, using conventional optimization methods

Pattern Radiation diagram
+
Antenna element weights

Train CNN

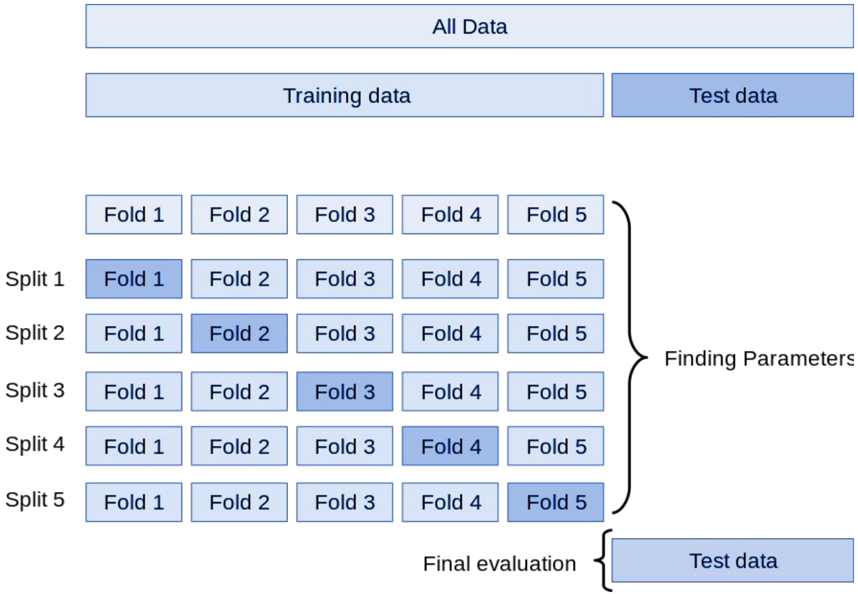
Testing I

The remaining 10% of the images will be used to test the network after the training. Using new patterns helps to understand and evaluate the learning process. After applying the output weights in the array, the obtained pattern will be compared with the initial one.



Testing II

The testing of the network can be further improved by using cross-validation, a technique that performs several learning cycles by taking to each one, a different portion of data for testing and training.



Pros & Cons

Pros

- The deep learning network can generate pattern synthesis weights much faster;
- During surveys, the array configuration can be quickly changed to focus on a specific area.

Cons

- Astrophysics works in static RFI environments;
- The network requires a large amount of data to train;
- The network is specific to a specific array geometry. Therefore, if the architecture of the array changes, the network needs to be retrained.

Future works

- Replace images with more significant features;
- Generate patterns using more advanced tools;
- Extend the functionality of the system also for multibeam patterns.

Thanks

Alessandro Cabras
Software Engineer @ INAF-OAC
alessandro.cabras@inaf.it