Machine learning for pulsar classification

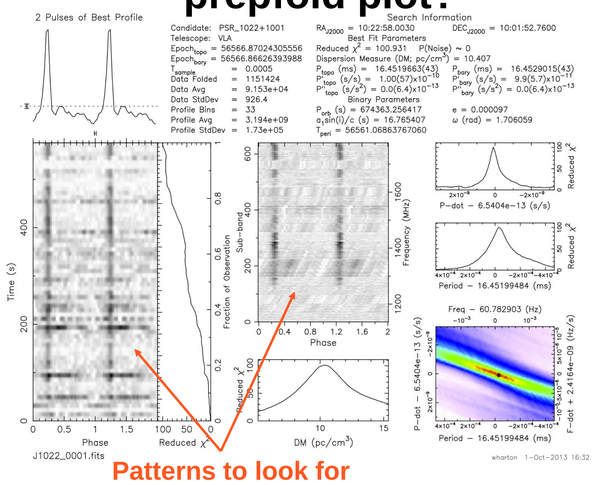
By: Dmytro Suprun

Advisors: Drs. Ann and Carl Schmiedekamp

Introduction

Gravitational waves from super-massive black hole binaries could be detected by pulsar timing arrays. To get the needed precision more new pulsars are required. Currently, pulsar candidates are found by manually inspecting thousands of plots from survey observations. Among those thousands of plots only a few are actual pulsars, the rest are just radio noise, known as RFI. This fact has caught my attention. As my results show, it is most certainly possible to train an artificial neural network to reject unlikely plots, which can greatly increase the likelihood of finding pulsar plots in a dataset.

What is a structure of the prepfold plot?



Patterns to look for

A prepfold plot is diagram, produced from the radio data, used to identify pulsars.

The diagram shows a likely pulsar when:

- Time vs Phase has vertical lines
- Frequency vs Phase has vertical lines
- DM peak > 0 pc/cm^3

How it's being done today

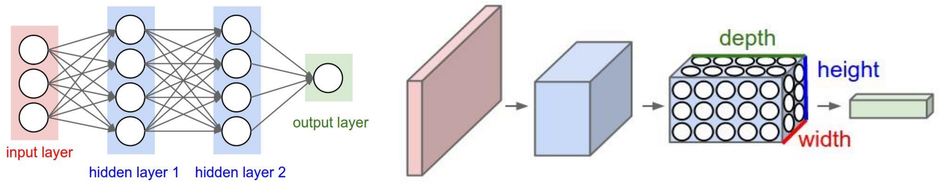
The process for identifying new pulsars is the following: survey data from Green Bank Telescope or Arecibo is processed and goes into creating plots, some known, definite RFI plots are deleted. there are hundreds of thousands of plots. Historically, about 1 in 10000 plots yields a new pulsar.

Our approach

1 out of 10000 means a lot of searching with very little expectation, and understandably, many students just give up. But what if finding a pulsar was 10 times more likely? Certainly, more would join in and more would continue searching. Additionally, deleting RFI would conserve space and potentially reduce the need for computing power, which is very cost-effective. Thus, it is only logical to use computers for this repetitive process. By using machine learning algorithms, we expect to eliminate many of the RFI plots, which will greatly increase percentage of true pulsar discoveries. The CNN[1] (convolutional neural network) will fit our needs as it is currently state-of-the-art in image recognition.

[1]Tensor Flow CNN; https://www.tensorflow.org/tutorials/deep_cnn

What is CNN ?

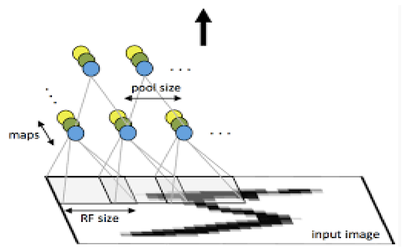


Regular neural network

Convolutional neural network

The convolutional neural network is known as shift invariant or space invariant artificial neural network.

Regular neural nets don't scale well to full images because of their fully connected structure. Thus, the advantage of CNN is that the neurons in a layer are only connected to a small region of the layer before it, instead of being connected to all of the neurons in the previous layer. This means that the CNN will not connect a background of a picture to an object in it.



Representation of locally connected neurons

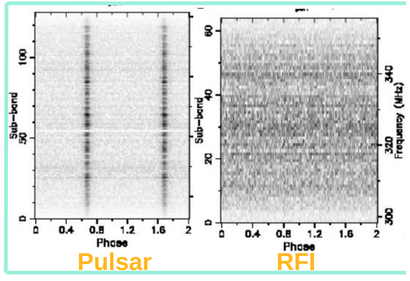
Setting up the environment

There is no need to make a CNN from scratch, instead it is possible to use transfer learning, which allows retraining a final layer of an already trained neural network. We used "Inception V3", which was trained on 1.2 millions pictures.

Sadly, there is no simple way to retrieve plots; they can be downloaded one at a time. For a million plots, it would require about 46 days. But we developed a Python program with the "mechanize" library for web-page manipulation, which reduced the download task to hours.

Initially, we trained the model on 2 categories, "Pulsars" and "RFI". We used 200 plots for training each category. On the right you can see illustrative examples of the two categories of plots.

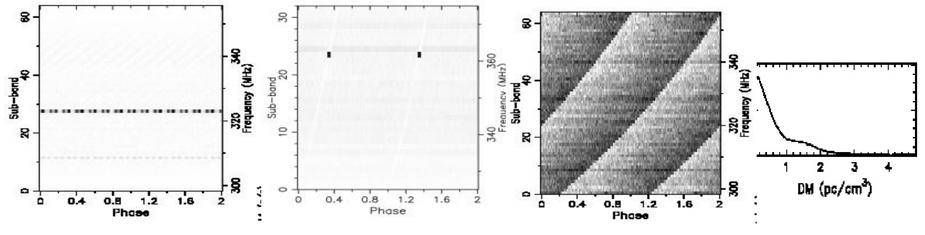
The Neural network outputs the probability of being in a category. It describes how much a given picture corresponds to a training category. Our initial model's accuracy was very poor. During testing our neural network would output a probability of 55 to 60 % for a pulsar plot, which is not an acceptable confidence level.




```
profi@profi-VirtualBox:~/changed$ python p_ch.py wavy.jpg
W tensorflow/core/framework/op_def_util.cc:332] Op BatchNorm
.batch_normalization().
pulsar (score = 0.87591)
rfi (score = 0.12409)
```

Results

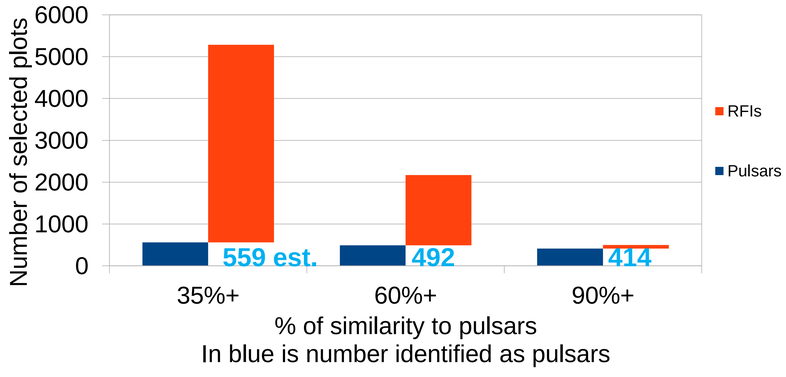
To increase accuracy, we introduced 4 more categories, examples of which you can see below. We accounted for plots with no vertical lines on one of the graphs, and for cases with wave-like graph forms. Also, we introduced a category for graphs with wrong DM(dispersion measure) curves.



After applying all those improvements, we succeeded in improving our neural net. We have provided the output of the neural net after it was asked to evaluate whether the plot was of a pulsar (the plot has not been shown to the neural net before). As you can see, our first neural net gave an output of 87.591% that the test image is a pulsar, but our second neural net gave 98% score that it is a wave-like plot. Indeed, it is a wave-like plot. So here we also illustrate another point: quantity and diversity is the name of the game, by increasing the amount of training data, and further diversifying data, we achieved much greater performance.

```
profi@profi-VirtualBox:~/changed$ python p_ch_3.py wavy.jpg
W tensorflow/core/framework/op_def_util.cc:332] Op BatchNormWi
.batch_normalization().
wave (score = 0.98066)
rfi (score = 0.01405)
pulsar (score = 0.00394)
baddm (score = 0.00095)
vstr (score = 0.00030)
novline (score = 0.00010)
```

After the training, we downloaded 98,000 plots from pulsar.wvu.edu. 5,284 of which were classified as 35%+ pulsars, rest are of different types of RFI. Thus, we decreased the amount of data by 19 times. We have identified more than 30 unique pulsars, and even though all of them are already known, the fact that we found all of them in just a couple of hours is mind boggling. We further proceeded by changing the lower bound of probability to 60% and 90%. The results are shown in histogram below.



This histogram represents the following trade off: as we increase the minimum pulsar acceptance value (35%, 60%, 90%) to filter out more RFI, we also lose some pulsars.

Conclusion

We can now apply our neural net for it's design purpose. As the results indicate, it will allow us to delete most of the RFI plots. Thus, our task is indeed achieved. But is it as good as it can get? No, it is not. Over the long testing period, we have seen it fail to detect a pulsar, while being 90% confident, that it is RFI. Why so? We think that lines on graphs might be so faint, that our neural net just doesn't see them. To further improve the accuracy, we are in a process of making our own neural net, which would take the raw arrays of intensity bins, without even converting them to plots.

Acknowledgments

We acknowledge the assistance of Prof. Maura Mclaughlin of West Virginia University, Sue Ann Heatherly, and the staff of the Green Bank Observatory, Greenbank, WV in operating the telescopes. We thank the Pennsylvania Space Grant Consortium (NASA) for financial assistance.