A COMPARATIVE STUDY OF NEUROEVOLUTION IN FRB CLASSIFICATION

Hari Prasad SV*
Affiliation (Current) : Department of Physics
University of Cologne
Cologne, Germany
hsreekri@smail.uni-koeln.de

July 3, 2024

ABSTRACT

Fast radio bursts (FRBs) are brief astrophysical events characterized by pulses lasting milliseconds and flux densities ranging from 0.1 to 100 Jy. Classifying these events from raw intensity data involves techniques such as de-dispersion for trial DM (dispersion measure) values and filtering out radio frequency interference (RFI). Since the first prototype discovery in 2007 (Lorimer et al.), the number of identified FRB sources has steadily increased.Ground-based surveys like the CHIME (Canadian Hydrogen Intensity Mapping Experiment) project have recently generated vast amounts of data, necessitating efficient analysis pipelines. This work explores the use of Genetic Neural Network algorithms for classifying FRB data and other raw data from diverse astrophysical surveys. The complexity of FRB data makes it ideal for investigating data classification in astrophysical contexts. Evolutionary algorithms offer parallelizability and faster runtimes, potentially enhancing analysis efficiency. This novel approach addresses the challenges associated with FRB data classification by employing Genetic Neural Network algorithms.

Keywords Fast radio bursts · Artificial neural networks · Neuroevolution

1 Introduction

Fast radio bursts (FRBs) are enigmatic astronomical transient phenomena of millisecond duration and flux densities in the range of $0.1 - 100$ Jy. *FRB*s belong to a much broader of class of *fast radio transients*, commonly referring to describe millisecond duration pulses that are produced by a coherent form of non thermal radio emission. The search for *fast radio transients* began in the late 1960s. With the discovery of the first Pulsar (Pulsating radio source) in 1967[1]. Pulsars exhibited a periodic behaviour, which from the analysis of the flux densities revealed a sharp rotation period. The origin of Pulsars was then determined to be the radio emissions from highly magnetized rotating neutron stars.

The development and expansion of radio surveys enhanced the search for fast radio transients. The open source availability of radio data also led to the widespread analysis and thus leading to the discovery of *Rotating Radio transients* (*RRATs*); highly intermittent galactic pulsars detected in single pulse searches. and *Giant Pulses* (*GPs*), pulses of extragalactic origin with highly intense individual pulses[2].

The existing searches eventually led upto the discovery of the first prototype *FRB* signature by *Lorimer et al.* while analysing the archival data from the *Megallanic clouds*[3]. Further four other sources were discovered by Thronton et al,[4]. The main characteristic of these signatures was the high frequency dispersion which cannot come from the Milky Way alone. Thus proving to be extra-galactic in origin.

A number of FRBs has been detected ever since, A web catalogue by Petroff et al.[5][2], contains FRB detections up to June 2020. In 2021 CHIME released a catalogue of 536 new FRBs observed from the CHIME survey.[6] .

*Implementations of the code that is used to generate the results in this article can be found in https://github.com/Cup-cake-lover
[2]www.frbcat.org

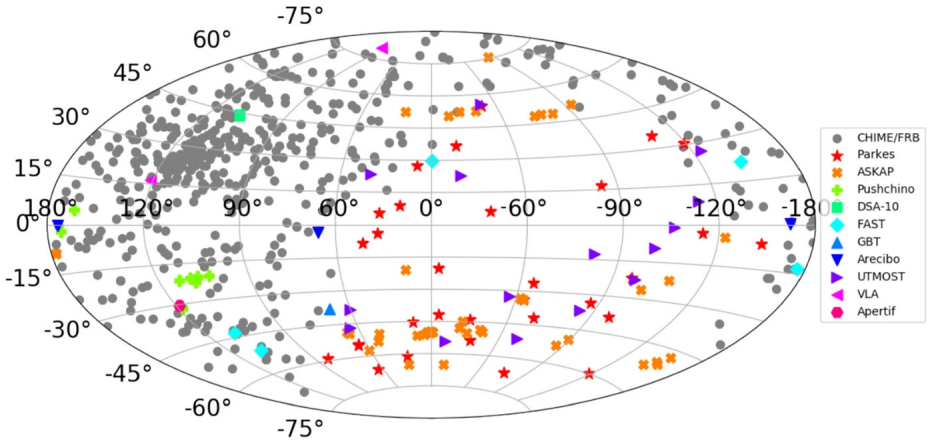


Figure 1: Distribution of observed FRBs till 2020 by Petroff et al.

CHIME's new 1024 beam survey has significantly enriched the detection rates. CHIME currently searches 400 - 800 MHz range, which facilitates detections of newer single pulse profiles.

2 Relevant terms and explanations

2.1 Interstellar medium

The interstellar medium (ISM) consists of dust and gas in the galaxy. As a radio wave propagates through ISM, it gets attenuated and undergoes other propagation effects.

2.2 Dispersion

The refractive index of the propagation medium; plasma is frequency dependent. This is called dispersion. The refractive index μ can be related to the frequency ν as,

$$\mu = \sqrt{\left(1 - \frac{\nu_p^2}{\nu^2}\right)} \tag{1}$$

Where the plasma frequency ν_p is directly dependent on the number density.

$$\nu_p = \sqrt{\frac{n_e e^2}{\pi m_e}}$$

where m_e, n_e, e are electron mass, number density and charge respectively. Due to dispersion, a time delay is observed between higher frequency waves and lower frequency waves.

$$t = \left(\int_0^d \frac{dl}{v_g}\right) - \frac{d}{c} \tag{2}$$

where d is the distance from earth. From this, the following integral can be obtained.

$$t = \frac{e^2}{2\pi m_e c \nu^2} \int_0^d n_e dl - \frac{d}{c} \tag{3}$$

The integral over the electron number density n_e over the distance from the source to observer is termed as the **Dispersion Measure** (DM). DM is measured in units of $\frac{pc}{cm^3}$. DM is used as a direct substitute for source distance measurement. Using DM and evaluating the expression, we can obtain the following relation relating the dispersion delay of two frequencies ν_1 and ν_2 as,

$$\Delta t = 4.15 \times 10^6 DM \left(\frac{1}{\nu_1^2} - \frac{1}{\nu_2^2} \right) \tag{4}$$

Different DM models are used for calculations, NE2001[7] and YMW16[8] models are widely used in galactic and intergalactic distance calculations.

2.3 Dynamic Spectra

Dynamic spectra provide valuable insights into the temporal morphological structure of radio events. They are typically represented as waterfall plots, which depict the variation of signal intensity over time and frequency. By plotting the intensity as a function of time and frequency, these plots capture the changing characteristics of the radio event. In the case of radio astronomy, dynamic spectra are particularly useful for studying FRBs.

When analyzing dynamic spectra, one can extract further information by dedispersing the data. Dedispersion involves taking the effective mean values along the columns of the frequency-time plot, resulting in a dedispersed pulse profile. This process helps to reveal the intrinsic pulse shape by compensating for the dispersion effects caused by the interstellar medium. While pulse profiles in some cases exhibit a Gaussian shape due to its sharp event detection, FRBs are known to exhibit a wide variety of pulse profiles, showcasing their diverse nature. A waterfall plot and its corresponding dedispersed pulse profile obtained from the CHIME catalogue[6].is shown below.

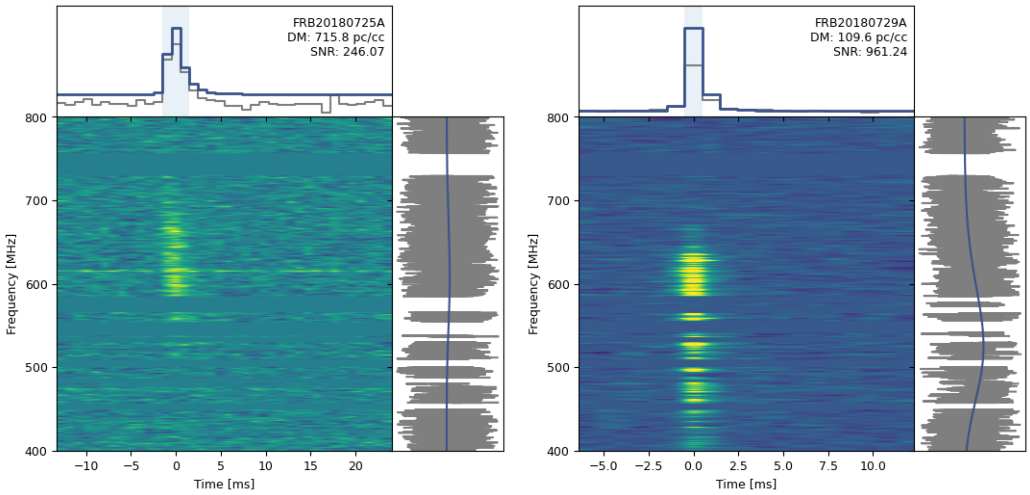


Figure 2: FRB20180725A & FRB20180729A - Dynamic Spectra along with dedispered pulse profile[6]

3 Brief overview of detection pipelines

Before delving into the classification, a brief understanding of the modern single pulse classification pipelines is required. A summary is given below[9]. These pipelines are regularly used in FRB surveys to detect and classify FRBs according to their various features (repeating, non repeating, DM etc)

3.1 Collection of radio data

The radio data is collected as raw voltage data. This analog data is converted into digital format by using analog to digital converters. This is then channelized using a spectrometer.[10] This channelized data is generally known as a filterbank.

3.2 Mitigation of RFI

The obtained data is often contaminated with radio noise from background, major sources include GPS, interference with broadcasting systems, routers etc. To minimize the effect of RFI, multiple techniques are utilized. Two most commonly used techniques are as follows.

- Thresholding: Reduction of RFI based on the assumption that the noise follows Gaussian noise profiles
- Zero DM subtraction: Mitigation of RFI based on appearance in all channels with zero DM due to lack of dispersion.

3.3 Dedispersion

The radio signals are dispersed due to the interaction with the ISM. To search for FRBs this effect due to dispersion needs to be removed. For doing this the data is analyzed using many many trial DM values. Since this method is practically a brute force, in single pulse searches, this step is computationally heavy and time consuming.

3.4 Normalization

After dedispersion, the data is smoothened over a window, by subtracting its mean. This smoothened data is then divided using the standard deviation which provides a normalized dataset. The amplitude of each sample becomes the **Signal to Noise ratio (SN)** of the sample.

3.5 Matched Filtering

Once the dedispersed data is acquired, it undergoes convolution with a boxcar kernel of varying widths to search for pulses. If a specific threshold signal-to-noise (SN) value is reached, it is marked as a potential candidate and forwarded for human examination.

4 Current challenges in Single pulse searches

Now that a general idea of the search pipeline is established, some existing challenges in detection pipelines are as follows.

1. Due to the relatively low field of view of current FRB surveys the number of detected events is low, even though there are thousands of likely events.
2. CHIME survey with 1024 beams is continuously searching for newer candidates in the range $400 - 800 MHz$ with very high DMs.[6] This increases the chances of detections.
3. But along with newer detections, the number of false positives above the threshold SN values will also increase. Hence the human inspection of all candidates will become intractable.
4. This calls for the effective use of automation in detection pipelines, instead of constant human inspections.

For automating the false positive detection process, machine learning can be effectively used .Previously effective classification using Convolutional neural networks with promising results has been done by multiple groups, including the baseline model used in this study by Connor et al. .[11]A detailed review of all the previous attempts are compiled by Agarwal et al. and can be found here[12]

This paper focuses on reviewing and comparing the pre-exisiting CNN classifier architectures trained with common stochastic gradient (SGD) methods with a much more robust Neuroevolutionary (NE) optimisation, this method in case of FRB classification has not been explored yet to best knowledge of the author.

5 A brief primer on Neuroevolution (NE)

The term 'Neuroevolution' (NE) can be split into constituent identifiers. where the term *Neuro* corresponds to something related to neural networks and *Evolution* related to the evolutionary algorithm used. Both will be briefly discussed below.

5.1 Artificial Neural networks[13]

Artificial neural networks are synthetic mathematical models that emulate biological neural systems. ANNs usually consist of multiple layers of 'neurons'. A MLP network (multi layer perceptron) is made by interconnecting these neurons and layers together in various ways. Each neuron computes a linear weighted sum of the inputs along with additional biases.

$$z = \sum_i w_i y_i + b_i \tag{5}$$

This value is fed to a non-linear function $\mathcal{F}(z)$. Usage of a non linear function is crucial since otherwise the output will be just linear function of the input. In deep learning, a number of activation functions are used. Some examples include,

- Sigmoid/Logistic Function

$$\mathcal{F}(x) = \frac{e^x}{e^x + 1} \tag{6}$$

- Rectified linear unit (ReLU)

$$\mathcal{F}(x) = max\,(0, x) \tag{7}$$

- Hyperbolic tangent (tanh)

$$\mathcal{F}(x) = tanh(x) \tag{8}$$

A deep neural network (DNN) consists of several of these neurons stacked on top of each other as layers. It typically contains a number of hidden layers. The hidden layers in which all neurons connected to each other are called fully connected or dense layers.

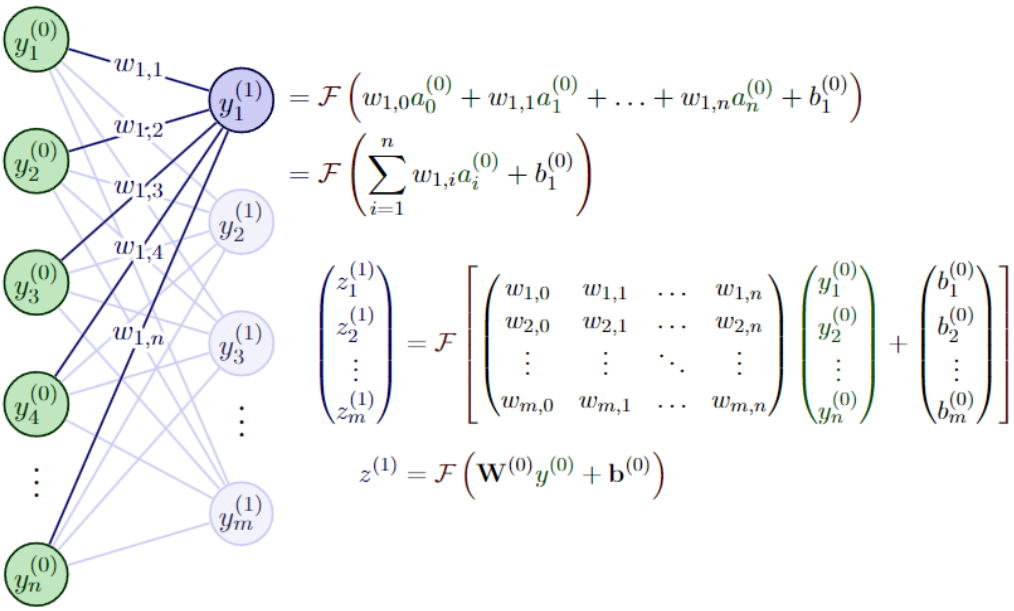


Figure 3: Schematic representation of a neuron activation

5.2 Training of a neural network - Backpropagation

Given the learning is supervised ie, the network has access to labels to match while training. The weights and biases needs to optimized such that it performs well at the given task.To optimize the weight matrix, first requirement is some large amounts of data that network can be trained with, this data is termed as '*Training data*' or '*Training set*'.

First the network is initialized with a random sets of weights and biases. Now the output Z is obtained by applying the operation mentioned in Eq. 5. To see how much is the difference between the actual output $\mathbf{Z}$ from the input $\mathbf{Y}$ A loss function $\mathcal{F}$ can be defined, which is just the mean squared error (MSE).

$$\mathcal{F}(\mathbf{W}, \mathbf{B}) = \frac{1}{N}\sum (z_i - y_i)^2 \tag{9}$$

where N is the number of samples, and $\mathbf{W}$ and $\mathbf{B}$ are the weights and biases corresponding to the loss function.

5.3 Minimizing the loss function

To minimize the loss and further optimize the weights, different strategies are employed. Two of them are discussed below.

5.3.1 Stochastic Gradient Descent

This method is generally used for minimising the loss function. To see this, first the loss function mentioned in Eq.9 is expanded in taylor series to the first order.

$$\mathcal{F}(\mathbf{W}+\Delta\mathbf{W},\mathbf{B}+\Delta\mathbf{B}) \approx \mathcal{F}+\frac{\partial\mathcal{F}}{\partial\mathbf{W}}(\Delta\mathbf{W})+\frac{\partial\mathcal{F}}{\partial\mathbf{B}}(\Delta\mathbf{B}) = \mathcal{F}+\nabla^T\mathcal{F}\Delta\mathbf{W}\Delta\mathbf{B} \tag{10}$$

To minimize the loss function, the term $\nabla^T\mathcal{F}\Delta\mathbf{W}\Delta\mathbf{B}$ is minimized. This parameter is updated accordingly, by defining a training parameter η such that ;

$$\kappa \leftarrow \kappa - \eta\mathcal{F}(\kappa) \tag{11}$$

Where $\kappa = \Delta\mathbf{W}\Delta\mathbf{B}$

The SGD method uses small steps iteratively for convergence. This method is generally termed as *Backpropagation*. SGD is regularly termed as an 'optimizer', since it optimises the parameter matrix. More generally, the values of the number of neurons, the weights, activation function which constitute a neural networks are called *Hyperparameters*. The tuning of these hyperparameters involves various techniques which is beyond the scope of this thesis.

5.4 Evolutionary algorithms

Evolutionary algorithms are set of heuristic algorithms,which is to mimic biological evolutionary processes. Some examples include Particle swarm optimisation (PSO), Differential evolution (DE), Ant colony optimisation (ACO), etc. A discussion on these algorithms is beyond the scope of this paper, a detailed overview can be found here. [14]

5.5 Genetic algorithms

Genetic algorithm (GA)is a meta heuristic algorithm inspired from the principle of Darwinian evolution.[15] Nature selects the most fittest individual by testing the adaptability of an individual organism to the environment itself. The algorithm was theorized by John Holland.[16] as an optimization method in the early 1980s. The algorithm tries to find the best possible solution from a number of solutions provided. The evolution is based on the idea of creation of parents, their traits are crossed over creating a daughter population, then the best suited individual from this pool is selected.

5.5.1 Genetic operators

After the parent pool is generated for the solution of a task, different genetic operations are imposed on it through an iterative process. These genetic operations are given below.[17]

1. **Selection:** Selection of a parent is dependant on the '*Fitness*' of the parent. A fitness is defined as a numerical score that is given for that solution for completing the task. Fitness are defined accordingly to the task at hand. A higher fitness score is always preferred for the selection of an individual.
2. **Crossover:** After certain number of parents are selected, the parents are allowed to '*exchange*' favourable features that they have. This directly corresponds to the crossover that is undergone in biological species. This ensures every generation produces better individual with a new genetic code.
3. **Mutation:** Even though crossover produces new individuals with better architecture, this doesn't necessarily lead to a converging solution, often the daughter pool can get stuck with similar type of solutions, to prevent this, and ensure genetic variety, random mutations in the genetic code are introduced. This produces exclusively new type of individuals, and ensures that solutions doesn't diverge.

Steps $1-3$ are repeated until a convergent solution is obtained.

The algorithm mentioned in the previous section is extremely efficient and is better adapted to noise due to its structure[18]. A striking feature of GAs are they can be implemented to optimize any type of population. This is where the idea of neuroevolution originally theorized.

Neuroevolution uses GAs as a backend to generate populations of neural networks that can be specifically designed for a task.

Evolution to ANNs have been introduced in multiple levels. Since the GAs have been so versatile on learning using different types of entry types (genotypes), provided that they undergo the genetic operations mentioned in subsection 5.5.1. These features can be introduced in :

- Connected weights and biases
- Overall architecture (NEAT).
- Learning Rules.

Encoding the information for the GA to process also becomes significant in this case, Similar to what mentioned in subsection 5.5.1.

5.5.2 Optimising connected weights and biases

This thesis tries to explore the simplest kind of neuroevolution, where the connected weights and biases are optimized. Before being fed into the genetic algorithm, the neural network needs to be encoded. A naive way of doing this is to encode the network as an adjacency matrix.[17] Consider a classic layered network with input layer with 2 neurons, one hidden layer with 3 neurons and an output later with one neuron. For simplicity the weights are assumed to binary, where connection between two neurons means a weight of 1 otherwise, 0. The adjacency matrix of this fully connected network is given below.

$$\begin{pmatrix} 0 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 & 1 & 0 \\ 1 & 1 & 0 & 0 & 0 & 1 \\ 1 & 1 & 0 & 0 & 0 & 1 \\ 1 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 & 1 & 0 \end{pmatrix}$$

To illustrate the neuroevolution process in more detail, consider a simple example. Let A and B be two parent neural networks with identical structures but with different sets of binary weights. ie,

$$A = \begin{bmatrix} 0 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 & 1 & 0 \\ 1 & 1 & 0 & 0 & 0 & 1 \\ 1 & 1 & 0 & 0 & 0 & 1 \\ 1 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 & 1 & 0 \end{bmatrix}; B = \begin{bmatrix} 0 & 0 & 2 & 2 & 2 & 0 \\ 0 & 0 & 2 & 2 & 2 & 0 \\ 2 & 2 & 0 & 0 & 0 & 2 \\ 2 & 2 & 0 & 0 & 0 & 2 \\ 2 & 2 & 0 & 0 & 0 & 2 \\ 0 & 0 & 2 & 2 & 2 & 0 \end{bmatrix}$$

According to subsection 5.5.1, These parent networks should now undergo a crossover. In this case, we assume a random crossover.

$$RandCross(A, B) = C \implies \begin{bmatrix} 0 & 0 & 2 & 2 & 2 & 0 \\ 0 & 0 & 1 & 1 & 1 & 0 \\ 1 & 1 & 0 & 0 & 0 & 1 \\ 1 & 1 & 0 & 0 & 0 & 1 \\ 2 & 2 & 0 & 0 & 0 & 2 \\ 0 & 0 & 2 & 2 & 2 & 0 \end{bmatrix}$$

Introducing mutation also yields,

$$Mutate(C) = D \implies \begin{bmatrix} 0 & 0 & 2 & 2 & 3 & 0 \\ 0 & 0 & 1 & 1 & 1 & 0 \\ 2 & 2 & 0 & 0 & 0 & 1 \\ 2 & 2 & 0 & 0 & 0 & 1 \\ 2 & 2 & 0 & 0 & 0 & 2 \\ 0 & 0 & 2 & 2 & 2 & 0 \end{bmatrix}$$

Now matrix D will be the final daughter matrix, and the fitness of this network representation is evaluated. And the process continues till a convergent solution is obtained.

6 Training dataset

In order to test and set some benchmarks, a valid dataset that is well understood needs to be used. For the current implementation a simulated dataset was considered due to the following reasons.

- Control over SNR (Signal to Noise ratio) of the candidates.
- Number of training candidates .
- Elimination of the additional post processing of the data (RFI mitigation and Uniformity in data shape).

The syntheis of the dataset is pretty straight forward, since there already exists a framework which can simulate single pulse candidates, within neccessary parameter constraints. The framework used here is *Single-pulse-ml*, developed by Connor et al.[3]. Using this framework, three sets of datatypes are obtained.

.

6.1 Frequency-Time Data

Frequency time data, or the dynamic spectrum, provides valuable insight to the temporal and morphological structure of transient events. They are represented as waterfall plots, which shows variation of signal intensity with respect to time and frequency. A sample generated using the framework is shown in Fig 4.

6.2 DM- Time data

The DM dedispersed data will be shown as a clump of DM data points presented as 'bow tie' pattern due to the degeneracy between optimal dispersion measure and pulse arrival time

6.3 Pulse Profiles

The Frequency-time data can be collapsed by averaging over columnar array indices and produces a 1D timeseries data that maximizes the SNR.

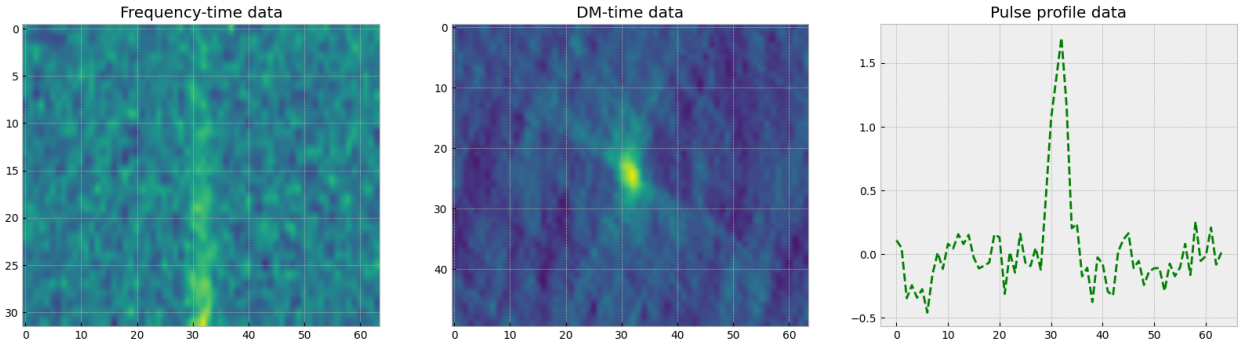


Figure 4: Figure showing Frequency-time, DM-time as well as didespersed pulse profile data. Synthesized by Single-pulse-ml framework

7 Implementation and Architecture

To compare Stochastic gradient descent and Neuroevolutionary backend a CNN neural architecture template is first decided and then trained by both backends for evaluation. The architecture is shown below. A 2D CNN based on Connor et al.[11] is used. For classifying 1D pulse profile data, a 1D CNN architecture is also designed.

[3] https://github.com/liamconnor/single_pulse_ml.git

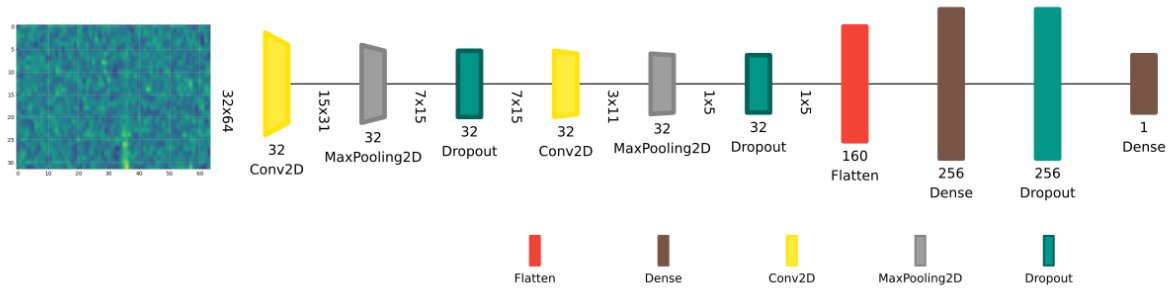


Figure 5: Frequency-time (and DM-time) 2D CNN template

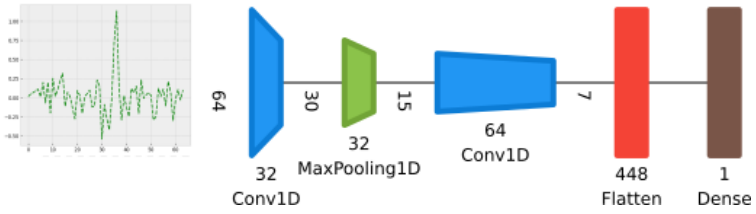


Figure 6: Pulse profiled 1D CNN template

The SGD backend was tested using Tensorflow Keras[19] sequential API and the NE backend was implemented using both Keras functional API as well as PyGAD[20], a python package developed for implementing Neuroevolution and genetic programming.

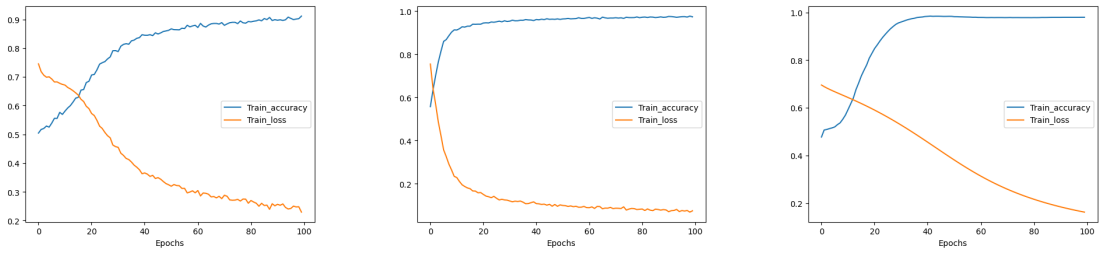
8 Comparison results between SGD and NE backends

The dataset mentioned in Section 6 are fed into specifically designed CNNs, these CNNs are designed accordingly to take in both 2 dimensional (Frequency-time and DM-time data) as well as 1D (Single pulse profile data). To compare both the classical stochastic gradient descent backend (SGD) and Neuroevolutionary backend(s) (NE), 3 evaluation metrics are chosen. These evaluation metrics are based on the parameters obtained from the confusion matrix as defined in Tab 1.

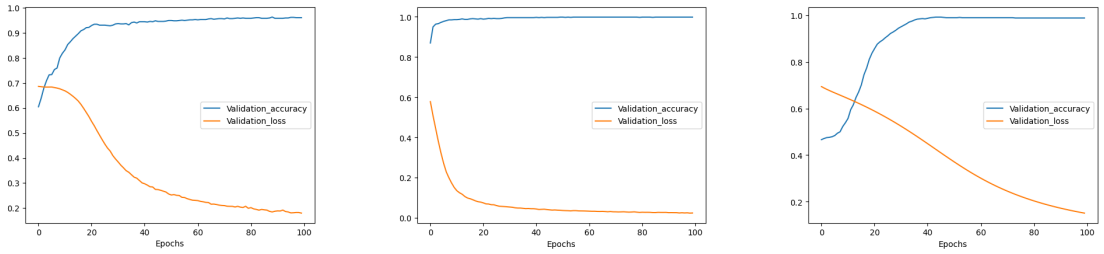
The training data was split into 80% training 20% testing. While training, 1% split was made in the initial training split for validation testing. The models were trained for 100 epochs with a batch size of 50.

Metric	**Formula**
Accuracy	$\frac{TP+TN}{TP+TN+FP+FN}$
Precision	$\frac{TP}{TP+FP}$
Recall	$\frac{TP}{TP+FN}$

Table 1: Metrics; Here TP, TN, FP, FN indicate True Positives, True Negatives, False Positives, False Negatives respectively.



(a) Frequency-time (b) DM-time (c) Pulse profiles



(d) Frequency-time (e) DM-time (f) Pulse profiles

Figure 7: Training stages with SGD backend: Figures (a) (b) (c) represent training with 100 epochs for Frequency-time, DM-time and Pulse profile data respectively. Figures (d) (e) (f) represent validation testing for Frequency-time, DM-time and Pulse profile data respectively

Metric	Frequency-time model	DM-time model	1D pulse profiles
Accuracy	96.85%	97.98%	98.20%
Precision	96.94%	99.11%	99.89%
Recall	96.94%	94.49%	96.44%

Table 2: Evaluation Metrics - Modified CNN - SGD

Where TP,TN,FP,FN are true positives, true negatives, false positives and false negatives respectively.

8.0.1 Genetic Algorithm backend

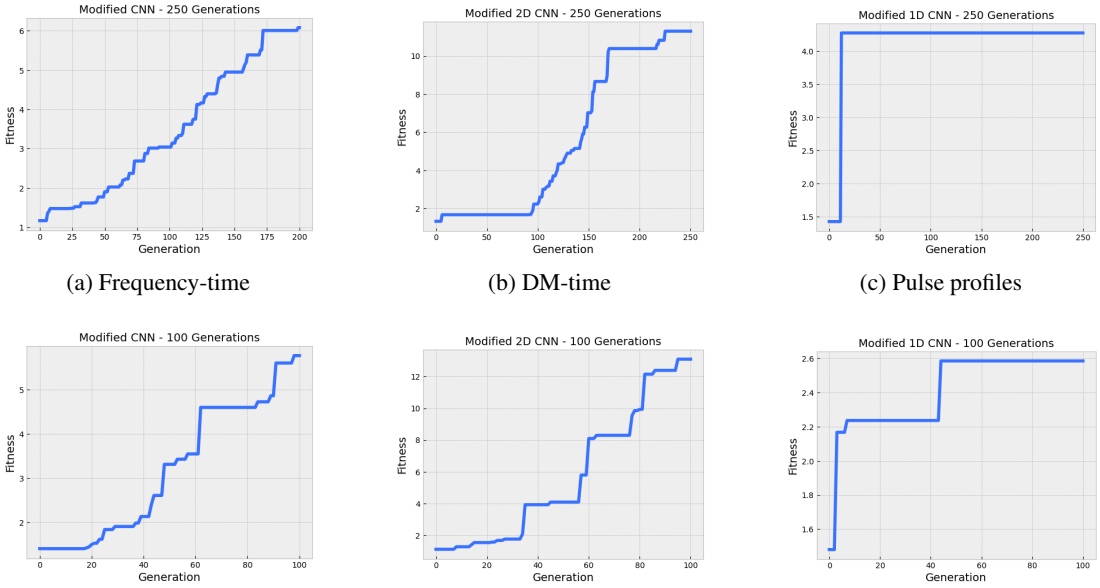
In this case, the genetic algorithm was allowed to run with the modified CNN as its seed for both 250 and 100 generations. In case of 250 generations, the number of parents that were mated was set to 5, while in case of the 100 generation run, the number of parents were reduced to two to observe possible differences.

Metric	250 Generations	100 Generations
Accuracy	95.60%	95.80%
Precision	98.06%	96.48%
Recall	92.88%	94.91%

Table 3: Evaluation Metrics for modified GA-CNN - Frequency-time Dataset

Metric	250 Generations	100 Generations
Accuracy	99.25%	98.85%
Precision	99.48%	99.79%
Recall	98.98%	97.86%

Table 4: Evaluation Metrics for modified GA-CNN - DM-time dataset



(a) Frequency-time (b) DM-time (c) Pulse profiles

(d) Frequency-time (e) DM-time (f) Pulse profiles

Figure 8: Training stages with NE backend: Fitness of the network v/s Generations are plotted, Figures (a) (b) (c) represent training with 250 generations and 5 parents for Frequency-time, DM-time and Pulse profile data respectively. Figures (d) (e) (f) represent training with 100 generations and 2 parents for Frequency-time, DM-time and Pulse profile data respectively

Metric	250 Generations	100 Generations
Accuracy	91.05%	93.00%
Precision	98.90%	94.42%
Recall	82.72%	91.15%

Table 5: Evaluation Metrics for GA-CNN - Pulse profile dataset

9 Discussion

From the results obtained for both SGD and GA optimisers, the accuracy values obtained were comparable and high for both cases. The most striking difference observed was the training process of both backends. Neuroevolution in this case was slower and was often stuck on a fitness score for a significant number of generations. But eventually due to randomness introduced by the mutation operators, The fitness score jumps. This is evidently seen in the fitness v/s generations plot.

Other factor that became crucial was the datatype used. Since the three types of datasets were used (Frequency - time, DM-time and Pulse profiles.) DM-time data gave significantly better results ($\approx 98\%$ accuracy). Hence one major change that is required for a seed architecture is the ability to use all three data types simultaneously. This will greatly improve the efficiency of the network architecture.

The GA based training was done for two sets of generations (250 and 100) with different number of parents (5 and 2 respectively). In case of all the training instances, the set with 2 parents that ran for 100 generations gave much better results. This was unexpected, since number of generations was supposed to constrain the fitness scores. But performed much better and faster.

In this particular test case, although stochastic gradient descent demonstrated comparable performance to the Genetic backend, there are some notable constraints associated with using the Genetic Algorithm (GA). These constraints include:

- Very simple neuroevolution scheme (Based on adjacency matrices)
- Constant mutation rates. (This limits the degree of change in the hyperparameters)

- Non-parallelized execution.

Neuroevolution is expected to perform better if the following changes can be made.

1. Instead of just optimising weights, entire architecture can be augmented while the training is done (NEAT)[21]. This completely ensures production of newer and more robust networks.
2. In the context of Fast Radio Burst (FRB) classification, the continuous expansion of the FRB database and the detection of new signal profiles necessitate constant modification of the classification network. Traditional convolutional neural networks (CNNs) often require the development of entirely new architectures to accommodate these changes. However, NEAT (NeuroEvolution for Augmenting Topologies) ensures that a classifier tailored to the specified requirements is always generated.

   By using NEAT, the classification network is continually refined and updated, enabling it to handle the ever-expanding FRB database effectively. The evolutionary nature of NEAT allows for the exploration of different network configurations and the selection of the most suitable ones based on performance and specification requirements. This adaptability ensures that the classifier remains up-to-date and capable of accurately classifying new FRB signals as they are detected.
3. Another possibility is the ability of the network to optimize learning rules. This will also increase the classification accuracy[22].
4. One big advantage of NE, which was not explored here, is parellelisation. The entire process of generation, fitness calculation mutation etc can be completely parllelized[23]. This is extremely crucial when such framework is integrated with existing search pipelines. The gives rise to the opportunity of generating a hundreds of parent networks and then their evaluation, which happens simultaneously. Which significantly increases the efficiency of search pipelines.

10 Conclusions and final remarks

In summary, the current study focuses on a preliminary exploration of neuroevolution and its potential application in FRB classification. The purpose is to establish a foundational framework that can serve as a starting point for further enhancements, incorporating the aforementioned remarks. By building upon this initial work, more advanced techniques can be developed, offering a competitive alternative to existing frameworks such as Generative Adversarial Networks (GANs)[24] in the field of FRB classification.

Acknowledgments

The author is grateful for the helpful review and guidance provided by Dr. Geetanjali Sethi and Dr. Annu Malhotra (Department of Physics,St. Stephens College, Delhi) and the Physics department of St. Stephen's College, Delhi for support.

References

[1] A. HEWISH, S. J. BELL, J. D. H. PILKINGTON, P. F. SCOTT, and R. A. COLLINS, "Observation of a rapidly pulsating radio source," *Nature*, vol. 217, pp. 709–713, 02 1968.

[2] R. N. Manchester, A. Lyne, F. Camilo, J. F. Bell, V. M. Kaspi, N. D'Amico, F. McKay, F. Crawford, I. H. Stairs, A. Possenti, M. Kramer, and D. Sheppard, "The parkes multi-beam pulsar survey - i. observing and data analysis systems, discovery and timing of 100 pulsars," *Monthly Notices of the Royal Astronomical Society*, vol. 328, pp. 17–35, 11 2001.

[3] D. R. Lorimer, M. Bailes, M. A. McLaughlin, D. J. Narkevic, and F. Crawford, "A bright millisecond radio burst of extragalactic origin," *Science*, vol. 318, pp. 777–780, 11 2007.

[4] D. Thornton, B. Stappers, M. Bailes, B. Barsdell, S. Bates, N. D. R. Bhat, M. Burgay, S. Burke-Spolaor, D. J. Champion, P. Coster, N. D'Amico, A. Jameson, S. Johnston, M. Keith, M. Kramer, L. Levin, S. Milia, C. Ng, A. Possenti, and W. van Straten, "A population of fast radio bursts at cosmological distances," *Science*, vol. 341, pp. 53–56, 07 2013.

[5] E. Petroff, L. Houben, K. Bannister, S. Burke-Spolaor, J. Cordes, H. Falcke, v. Haren, A. Karastergiou, M. Kramer, C. Law, v. Leeuwen, D. Lorimer, O. Martinez-Rubi, J. Rachen, L. Spitler, and A. Weltman, "Voevent standard for fast radio bursts," *arXiv e-prints*, p. arXiv:1710.08155, 10 2017.

[6] M. Amiri, B. C. Andersen, K. Bandura, S. Berger, M. Bhardwaj, M. M. Boyce, P. J. Boyle, C. Brar, D. Breitman, T. Cassanelli, P. Chawla, T. Chen, J.-F. Cliche, A. Cook, D. Cubranic, A. P. Curtin, M. Deng, M. Dobbs, F. (Adam) Dong, G. Eadie, M. Fandino, E. Fonseca, B. M. Gaensler, U. Giri, D. C. Good, M. Halpern, A. S. Hill, G. Hinshaw, A. Josephy, J. F. Kaczmarek, Z. Kader, J. W. Kania, V. M. Kaspi, T. L. Landecker, D. Lang, C. Leung, D. Li, H.-H. Lin, K. W. Masui, R. Mckinven, J. Mena-Parra, M. Merryfield, B. W. Meyers, D. Michilli, N. Milutinovic, A. Mirhosseini, M. Münchmeyer, A. Naidu, L. Newburgh, C. Ng, C. Patel, U.-L. Pen, E. Petroff, T. Pinsonneault-Marotte, Z. Pleunis, M. Rafiei-Ravandi, M. Rahman, S. M. Ransom, A. Renard, P. Sanghavi, P. Scholz, J. R. Shaw, K. Shin, S. R. Siegel, A. E. Sikora, S. Singh, K. M. Smith, I. Stairs, C. M. Tan, S. P. Tendulkar, K. Vanderlinde, H. Wang, D. Wulf, and A. V. Zwaniga, "The first chime/frb fast radio burst catalog," *The Astrophysical Journal Supplement Series*, vol. 257, p. 59, 12 2021.

[7] J. M. Cordes, "NE2001: A New Model for the Galactic Electron Density and its Fluctuations," in *Milky Way Surveys: The Structure and Evolution of our Galaxy* (D. Clemens, R. Shah, and T. Brainerd, eds.), vol. 317 of *Astronomical Society of the Pacific Conference Series*, p. 211, Dec. 2004.

[8] J. M. Yao, R. N. Manchester, and N. Wang, "A New Electron-density Model for Estimation of Pulsar and FRB Distances," *apj*, vol. 835, p. 29, Jan. 2017.

[9] E. Petroff, J. W. T. Hessels, and D. R. Lorimer, "Fast radio bursts," *The Astronomy and Astrophysics Review*, vol. 27, p. 4, May 2019.

[10] D. A. Roshi, M. Bloss, P. T. Brandt, S. Bussa, H. Chen, P. Demorest, G. Desvignes, T. Filiba, R. W. Fisher, J. B. Ford, D. T. Frayer, R. W. Garwood, S. Gowda, G. N. Jones, B. Mallard, J. D. Masters, R. McCullough, G. Molera, K. O'Neil, J. D. Ray, S. D. Scott, A. L. Shelton, A. Siemion, M. C. Wagner, G. Watts, D. Werthimer, and M. Whitehead, "Advanced multi-beam spectrometer for the green bank telescope," 10 2011.

[11] L. Connor and J. van Leeuwen, "Applying deep learning to fast radio burst classification," *The Astronomical Journal*, vol. 156, p. 256, 11 2018.

[12] D. Agarwal, "Searches for fast radio bursts using machine learning," 01 2020.

[13] N. Saha, A. Swetapadma, and M. Mondal, "A brief review on artificial neural network: Network structures and applications," in *2023 9th International Conference on Advanced Computing and Communication Systems (ICACCS)*, vol. 1, pp. 1974–1979, 2023.

[14] F. G. Mohammadi, A. M. Hadi, and H. R. Arabnia, "Evolutionary computation, optimization and learning algorithms for data science," *arXiv (Cornell University)*, 08 2019.

[15] S. Katoch, S. S. Chauhan, and V. Kumar, "A review on genetic algorithm: past, present, and future," *Multimedia Tools and Applications*, vol. 80, pp. 8091–8126, 2021.

[16] J. H. Holland, "Genetic algorithms," *Scientific American*, vol. 267, no. 1, p. 66–73, 1992.

[17] K.-C. Wong, "Evolutionary algorithms: Concepts, designs, and applications in bioinformatics: Evolutionary algorithms for bioinformatics," *CoRR*, vol. abs/1508.00468, 2015.

[18] T. Then and E. Chong, "Genetic algorithms in noisy environments," *ECE Technical Reports Electrical and Computer Engineering*, vol. 12, 1993.

[19] F. Chollet, "Keras," 2015.

[20] A. F. Gad, "Pygad: An intuitive genetic algorithm python library," 2021.

[21] K. O. Stanley and R. Miikkulainen, "Evolving neural networks through augmenting topologies," *Evolutionary Computation*, vol. 10, pp. 99–127, 06 2002.

[22] K. O. Stanley, "Compositional pattern producing networks: A novel abstraction of development," *Genetic Programming and Evolvable Machines*, vol. 8, pp. 131–162, 2007.

[23] J. Runwei Cheng and M. Gen, "Parallel genetic algorithms with gpu computing," *Industry 4.0 - Impact on Intelligent Logistics and Manufacturing*, 03 2020.

[24] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial networks," *arXiv (Cornell University)*, 06 2014.