# ARCHITECTURE

This is the architecture of the .ipynb file which includes the main code of setting up the model for classification.

|  |
| --- |
| Name- submission\_iitr.ipynb |
| Load Libraries |
| Define Important Functions |
| Load Data |
| Data Pre-processing |
| Model Building |
| Model Training |
| Model Analysis |
| Model Ensemble |
| Result |

This is the architecture of the files which constitute the Genetic Algorithm in the ‘genetic\_algorithm’ folder. Calling the main function by setting the parameters for ‘main’ and the model in ‘train’ can gives the hyperparameters best for the defined model.

|  |  |  |
| --- | --- | --- |
| Name – main.py | Name- Optimiser.py | Name- train.py |
| This file contains the main function which includes the number of population and the number of generations. | This files returns the value sets that are optimised for the score of the model. | This file contains the model to be trained and it also returns the score to the optimiser for optimisation. |

# METHODOLOGY

**Data Pre-processing**

The biggest hurdle in exoplanet detection is dealing with the noise in the data (either due to systematic error of the asteroids). Various smoothing techniques like the Gaussian filter, Total variance de noising, Rolling Mean, Rolling Median, Fourier transform were done. Fourier transform was done and frequencies beyond a certain percentile were eliminated as frequency of planetary motion will have less amplitude as compared to that of Random Noise or systematic error of device.

After smoothing the time series was stacked with the original time series and classification was done on this data as a comparison could help the model to analyse the local variation as well as the global variation in the time series.

A Dynamic filter was made by a skip connection in the neural network to allow the original time series to be stacked with the activations of the dynamic convolution filter. The details of this architecture is in the models block diagram.

**Neural Network Architecture.**

Convolutional neural networks (CNNs) exploit spatial structure of the intensity variation by learning local features. By adding a Maxpool Layer which aggregates values within small neighbourhoods, it also becomes invariant to small translations in the input, which is helpful as patterns in time series don’t have precise location in them. Further LSTM is well-suited to classify time series given time lags of unknown size as they can remember patterns over arbitrary time intervals.

Neural networks are strong functions which can overfit due to the large number of parameters. Dropout regularization are added to fully connected layers to help avoid overfitting on the sample space by randomly “dropping” some of the output neurons from each layer during training thus preventing the model from becoming overly reliant on any of its features. The activations of the neurons are made non-linear with the help of ‘relu’ activation function. Batch Normalisation at each step reduces the internal covariance shift and thus providing a support to the large number of neurons.

Hyper parameters of neural network are tricky as the search space is large. Genetic algorithm can improve the hyper parameter tuning by representing the possible sets of the parameters as individuals and repeatedly modifying them (by mutation, crossover) in successive generations alongside eliminating the weak individual. Block Diagram of Working of a Genetic algorithm is described as a flow chart.

The skip connection for the aforementioned dynamic smoothening is also shown in the diagram.

Model Architecture Printed in the Jupyter notebook due to lack of space

**Training Procedure.**

The data is split into train and test data (train: test: 10:1). To handle the imbalance in the train set (33 planet, 3927 non-planet) batches of data size containing 48 examples (16 planet, 32 non-planet) are generated on the fly (while training) in the batch\_generator function. Further the time series is rolled by a random seed to generate horizontal reflection of the series. This basically helps to augment the data by generating similar instances of the original data and thus helps the model learn by avoiding overfitting of the major class.

Adam optimiser algorithm with a learning rate of .001(35 epochs) is used to minimize the binary cross-entropy error function. The network is further fine-tuned with a leaning rate of .0001(10 epochs) and .00001(10 epochs) successively.

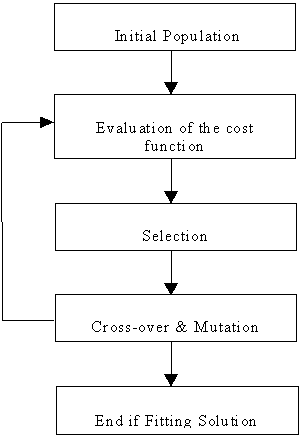
**Model Averaging**

Ensembles of two CNN models and an LSTM models.

# NOVELTY

The dynamic filter which works to stack a convolved output of the time series with the original data increased the f1 score significantly. This works as choosing one filter (fourier or gaussian1d) for all the data restricts the model to do the smoothening in a confined way rather than selecting different filters for different examples. Replacing the task of a filter with a convolution layer and making a skip connection generalised the model.

Hyper parameter tuning could be done with the help of grid search but it would make the search space very large (~45k). Training so many models is not feasible so Genetic Algorithm was implemented to reduce the search space. The algorithm begins by creating a random initial population and then selects 40 percent of the top performers. These 40 percent sets are propagated by crossover, mutation to increase the population of the selected class. The same process is repeated for n number of iteration. This model is inspired by the biological process of the survival of the fittest.



Block Diagram of working of a Genetic Algorithm.

As the data was imbalanced, it was augmented to create batches of data that helped in reaching a good score by overcoming the imbalance in train data. The time series classification was to identify patterns prevalent in the data and rolling it created a similar pattern which can be considered of the same class.

Several models had similar score but different false negatives and false positive examples. Making an ensemble of these models helped to increase the score as they captured the data differently.

# REFERENCES

https://lethain.com/genetic-algorithms-cool-name-damn-simple/

http://ieeexplore.ieee.org/document/6579659/?part=1

<https://www.whitman.edu/Documents/Academics/Mathematics/2014/carrjk.pd>

[Artificial Intelligence on the Final Frontier: Using Machine Learning to Find New Earths - Abraham Botros](https://pdfs.semanticscholar.org/8c96/1f7357fbc3e22e0e5417744af24662222818.pdf)

https://www.cfa.harvard.edu/~avanderb/kepler90i.pdf