Gravitational Waves Detection and Glitch Classification using Convolutional Neural Network

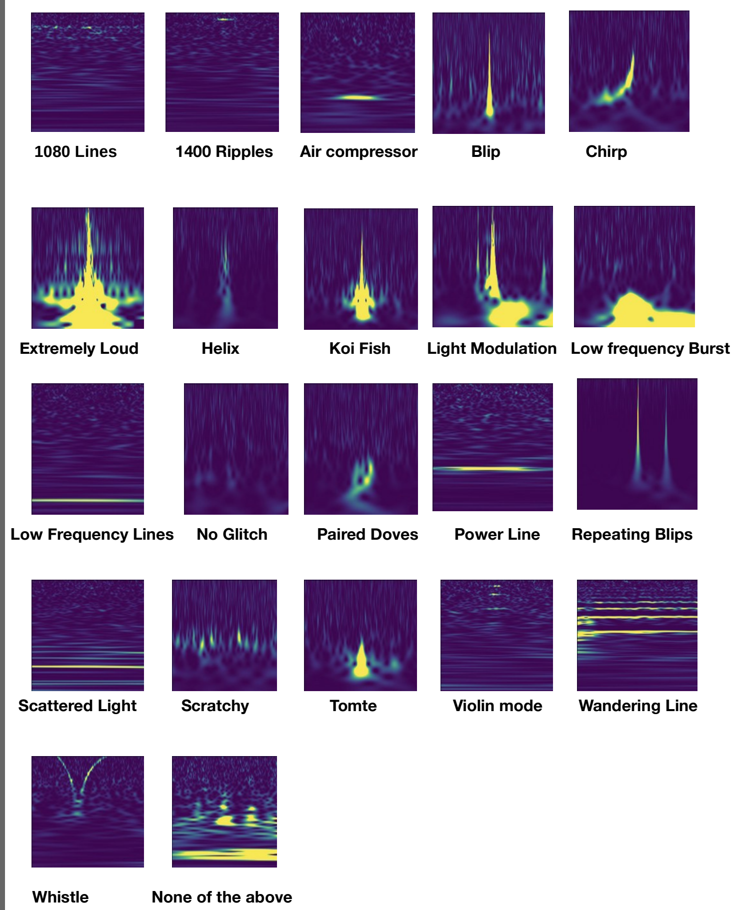
Anirban Bairagi
Department of Physics, Indian Institute of Technology, Kharagpur

Introduction

Gravitational Waves are ripples in space-time produced by the bulk accelerated motion of matter. Gravitational Waves were first predicted by Albert Einstein in 1916 as a consequence of his work on general relativity. The first indirect observational proof of the existence of Gravitational Waves was from a binary pulser PSR1913+16 in 1974 when Hulse and Taylor noticed that the time period of the pulser is decreasing. However, now it is possible to detect Gravitational Waves directly using Interferometric Gravitational Wave Detectors. When GW passes through the detector it causes strain in the detector arms and GW detector data records the strain observed at different times. Unfortunately the data contains a lot of noise. In most cases the noise is frequency dependent and non stationary and are produced by interactions among detector subsystems or with the surrounding environment. Transient noise glitches often originate from complex nonlinear couplings between the channels of the detector, thus resulting in very complex time-frequency morphologies. The characteristics of the noises may also depend on the quantum state of the gravitational field. Thus detection of this fundamental noise may provide some direct evidence for the quantization of gravity and the existence of gravitons. It is easier to visualise these noise morphologies in time-frequency domain (Spectrogram) of the data that allows us to select out features at different frequencies, and note how they evolve over very short times, without much prior knowledge of the signal morphology.

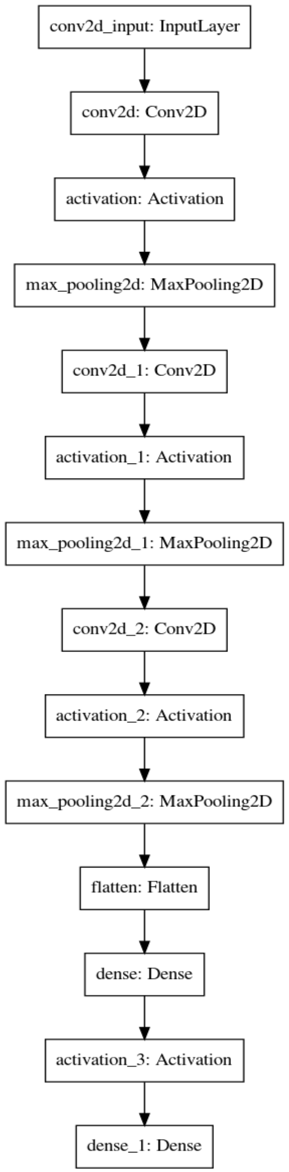
I have developed a Machine Learning (Convolutional Neural Network) model which can classify GW glitches of 22 classes from GW Spectrogram with an accuracy of approximately 95% after 3 hours of training. Once a glitch family has been identified, it is possible to perform further investigations to establish its origin and prepare custom data quality flags in order to reduce their impact on the detector performance. We can also detect any GW event very quickly without estimating parameters at first using this classification model, since a GW event produces Chirp-like spectrograms. Later we can use Matched Filtering for parameter estimation and for confirming the event. However, this takes much more time as it involves template banks for cross checking.

GW Glitch Classes



Methodology

CNN Model



Model

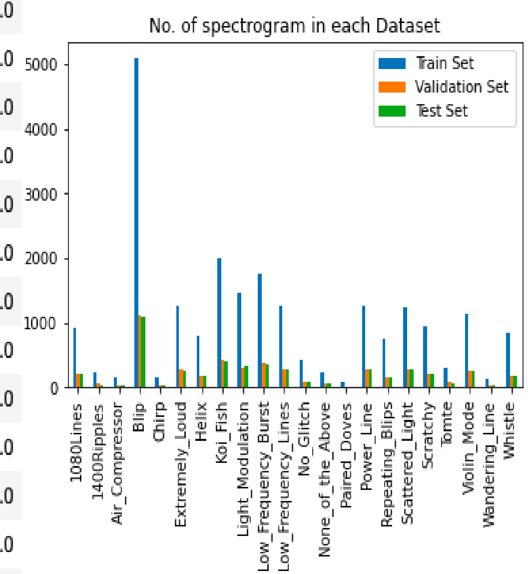
This machine learning model consists of three convolution operation each of which followed by ReLU activation and max pooling layer. These are done to downsample the image and extract important features. Then this image has been flattened followed by two dense layer to get the probabilities of the image belongs to each of the 22 classes. The position of the maximum probable number makes the prediction of which type of noise does it consists of among the 22 classes. Here I have used softmax cross entropy loss as cost function.

Input Dataset

This model takes labelled spectrogram images of different classes as input. I have used a Gravity Spy dataset. It contains Train, Validation and Test data directory and each of these folder contain spectrogram images of different classes. In this table we have listed number of input images of each classes and corresponding histogram has been shown for better visualisation. From the histogram for Train Dataset we can say there is class imbalance which may lead to greater overall loss during validation and testing.

Number of Spectrograms in Dataset

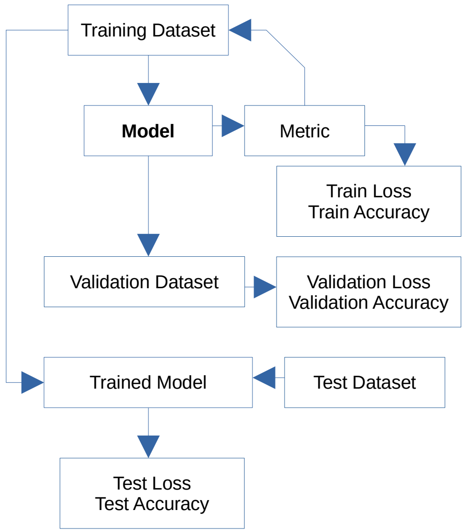
	Train Set	Validation Set	Test Set
1080Lines	916.0	196.0	200.0
1400Ripples	236.0	52.0	36.0
Air_Compressor	164.0	32.0	36.0
Blip	5096.0	1100.0	1092.0
Chirp	164.0	36.0	40.0
Extremely_Loud	1264.0	268.0	256.0
Helix	780.0	168.0	168.0
Koi_Fish	1992.0	424.0	408.0
Light_Modulation	1444.0	292.0	312.0
Low_Frequency_Burst	1748.0	376.0	360.0
Low_Frequency_Lines	1260.0	264.0	264.0
No_Glitch	428.0	88.0	84.0
None_of_the_Above	228.0	52.0	44.0
Paired_Doves	76.0	16.0	16.0
Power_Line	1256.0	268.0	272.0
Repeating_Blips	740.0	164.0	148.0
Scattered_Light	1232.0	272.0	268.0
Scratchy	948.0	200.0	200.0
Tomte	292.0	68.0	52.0
Violin_Mode	1136.0	256.0	256.0
Wandering_Line	116.0	24.0	28.0
Whistle	832.0	184.0	180.0



Application of CNN Model

GW Glitch Classification

At first the Train dataset is feed into the model. It gives loss and accuracy as output. Here loss is our metric and minimising the loss is our main target. Then we do back-propagation to update the values of the model parameters. In this way we train the model 20 times and at each step we observe the loss and accuracy on the Validation set. After 20 epochs the trained model is applied on Test dataset to obtain test loss and test accuracy for each classes.



Loss and Accuracy

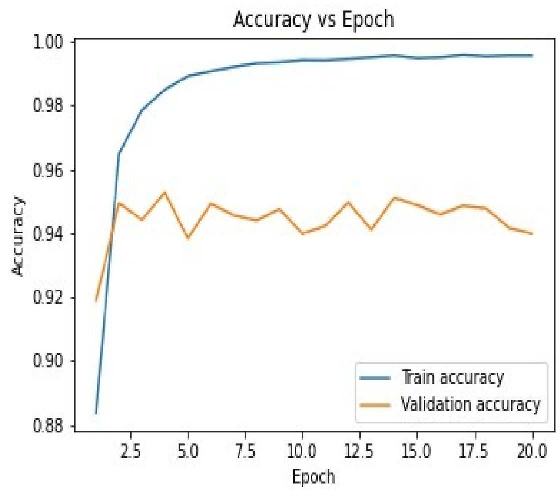
Here accuracy are given between 0 and 1 i.e 1 means 100% accuracy. After 20 epochs we obtain

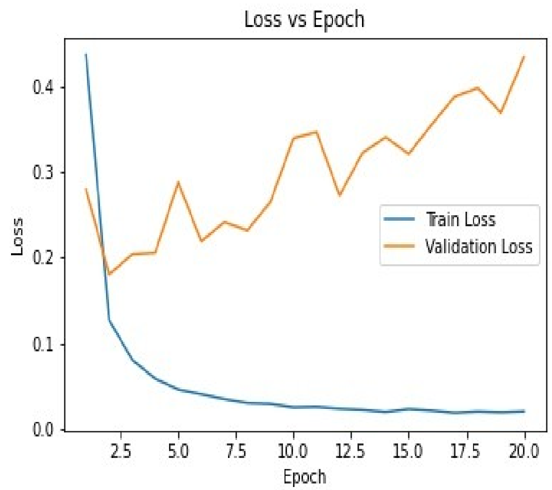
Train accuracy: 0.9956

Validation accuracy: 0.9398

Train loss: 0.02

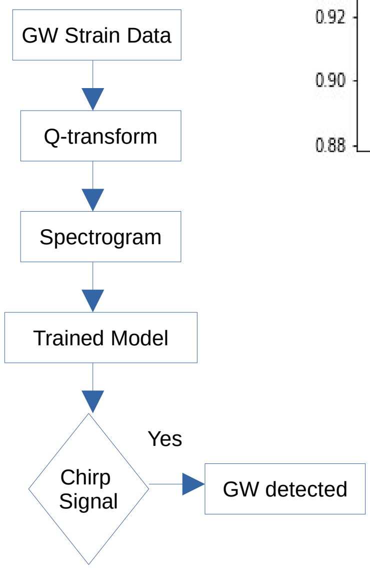
Validation loss: 0.434





GW Detection

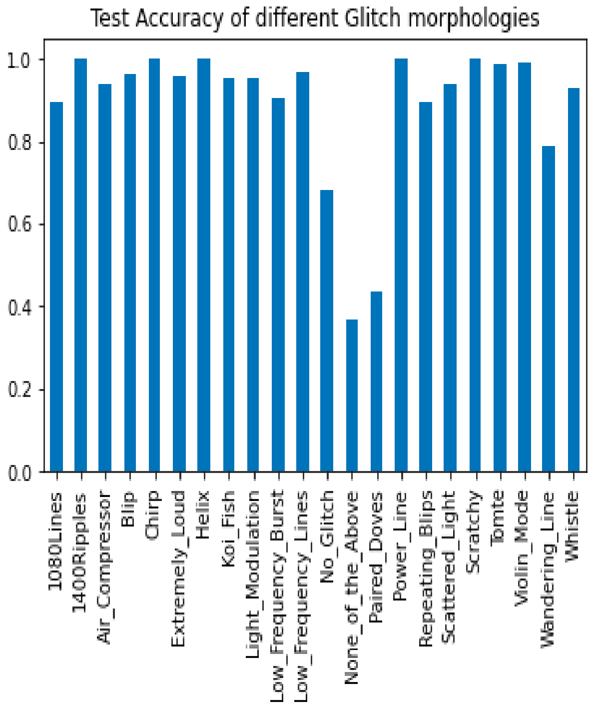
Taking a GW Detector Time-series data one can apply Q-transform to generate Spectrogram of the data. On running this trained model on the slices of the spectrogram it is possible to predict which type of noise does it belongs to. Since GW events create Chirp like spectrogram, if it finds any Chirp like signal observed will be notified.



Result

Test Loss and Accuracy

Test accuracy: 0.9423
Test loss: 0.522
The accuracy and loss of each class of Test Dataset has been shown in the Table and corresponding histogram.



	Test accuracy	Test loss
1080Lines	0.892857	1.092550e+00
1400Ripples	1.000000	2.173559e-06
Air_Compressor	0.937500	6.785514e-01
Blip	0.963393	4.874319e-01
Chirp	1.000000	2.345851e-03
Extremely_Loud	0.957031	6.317230e-01
Helix	1.000000	1.756487e-03
Koi_Fish	0.951122	2.068618e-01
Light_Modulation	0.952083	4.984309e-01
Low_Frequency_Burst	0.906250	6.957481e-01
Low_Frequency_Lines	0.968750	1.481155e-01
No_Glitch	0.683333	1.305593e+00
None_of_the_Above	0.369792	7.010175e+00
Paired_Doves	0.437500	2.431131e+00
Power_Line	1.000000	8.278422e-10
Repeating_Blips	0.896250	7.324809e-01
Scattered_Light	0.938657	1.134855e+00
Scratchy	1.000000	1.594712e-06
Tomte	0.984375	8.960066e-02
Violin_Mode	0.992188	1.379423e-02
Wandering_Line	0.785714	2.522329e+00
Whistle	0.929167	3.906871e-01

Conclusion

After 3 hours of training, the model gives overall accuracy of 94.23% and loss of 0.522 on Test Dataset. This model identify Chirp, 1400 Ripples, Helix, Power Line, Scratchy with 100% accuracy and most of the other classes with more than 95% accuracy. Those classes giving low accuracy is due to very less Train data from that class thus, gives relatively high loss. Resampling the Train set using data augmentation or adding other new data to reduce class imbalance will definitely help in reducing Test loss and increasing Test accuracy. From the accuracy vs epoch plot for Train and Validation set we can notice overfitting. Use of Dropout or other Regularisation technique will reduce overfitting effect. Since this model recognise the Chirp signal with 100% accuracy, we can detect Gravitational Waves from Spectrogram of a GW time-series data using this classification model very fast with a very good accuracy.

Due to lack of computational power, I have not been able to use deeper model, which may increase Test accuracy.

Acknowledgement : I would like to acknowledge Dr. Sayan Kar of Department of Physics, IIT Kharagpur for his constant support and guidance. I would also like to thank LSC for organising LIGO Open Data Workshop from where I have learnt a lot of useful things.

References:

1. M. Razzano and E. Cuoco, Classical Quantum Gravity 35, 095016 (2018) .
2. M. Parikh, F. Wilczek and G. Zahariade, The Noise of Gravitons, 2005.07211

If anyone has any query or would like to collaborate with me please feel free to contact me.

Email ID: **anirbanbairagi8@gmail.com**
anirbanbairagi@iitkgp.ac.in