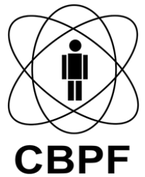
Big Data and Astroinformatics

Gravitational Wave Detection: Find gravitational wave signals from binary black hole collisions



Caroline Sodré (CBPF)
December, 2022

Sumário

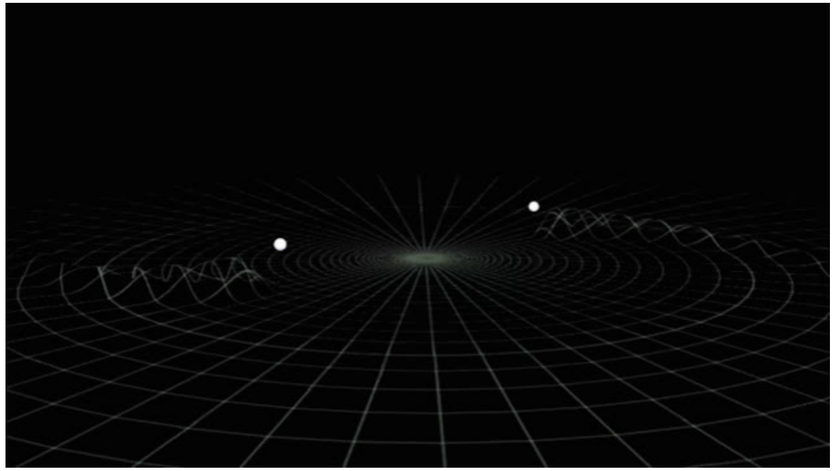
Objetivo

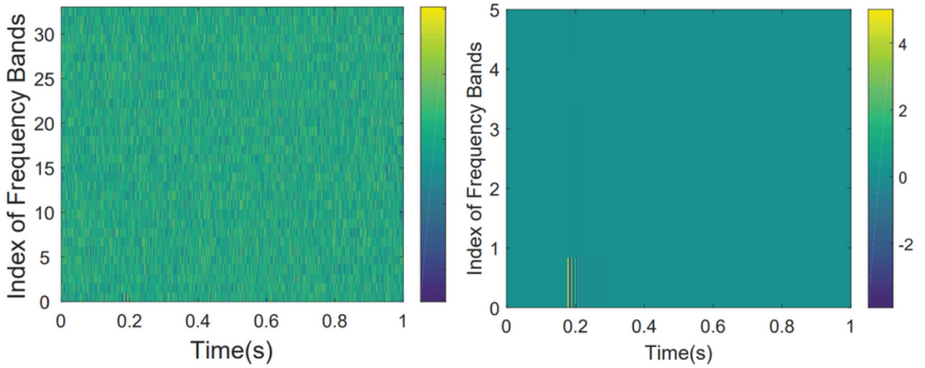
O objetivo é propor uma solução “simplificada”, porém, eficiente para o desafio “G2Net Gravitational Wave Detection: Find gravitational wave signals from binary black hole collisions” lançado no Kaggle

O problema consiste em calcular a probabilidade de um conjunto de medições simuladas de ondas gravitacionais de 3 interferômetros de ondas gravitacionais (LIGO Hanford, LIGO Livingston e Virgo) conter apenas ruído (target =0) ou conter ondas gravitacionais e ruído (target=1)

Introdução

- Ondas gravitacionais são "ondulações" invisíveis no tecido do espaço-tempo
- Albert Einstein, 1916
- LIGO, 2015
- Apesar de termos acesso a detectores extremamente sensíveis, os sinais acabam se misturando com ruídos do detector





Estratégia adotada

1

nnAudio.Spectrogram.CQT1992v2

Transformar os dados disponibilizados para o domínio da frequência

2

Keras + EfficientNet

Feature Extraction + Fine-Tuning

Feature Extraction em imagens

Processo de identificar as características mais importantes de imagens e transformá-las em features numéricos compactos que podem ser processados mais facilmente

Ajuda a reduzir efetivamente a quantidade de dados e aumenta a velocidade das etapas de aprendizado

Fine-Tuning

Técnica de Transfer Learning

Consiste em construir um modelo seguindo os seguintes passos:

1. Pegue camadas de um modelo previamente treinado excluindo o output layer e congele-as
2. Adicione um novo output layer sobre o modelo pré-treinado
3. Treine o novo modelo em seu conjunto de dado
4. Descongele todo o modelo (ou parte dele) e treine-o novamente nos novos dados

1 nnAudio.Spectrogram.CQT1992v2

O nnAudio é uma ferramenta PyTorch para processamento de áudio que pode calcular diferentes tipos de espectrogramas em tempo real

Sua função nnAudio.Spectrogram.CQT1992v2 calcula o Constant Q Transform (CQT) de um dado sinal mapendo-o do espaço do tempo para o espaço de frequência de modo que os bins de frequência sejam espaçados geometricamente

2 Keras

O Keras é um neural network API desenvolvido pelo Google para construir modelos de machine learning

Escrito em Python, seu objetivo é tornar mais fácil o processo de implementação de redes neurais

É considerado extremamente beginner-friendly

É incorporado ao TensorFlow e pode ser usado para realizar deep learning rapidamente, pois fornece módulos embutidos para todos os cálculos de redes neurais

3 EfficientNet

A família de modelos EfficientNet é um conjunto de redes neurais convolucionais do GoogleAI que podem ser usadas como base para uma variedade de tarefas de visuais, embora tenham sido inicialmente projetadas para classificação de imagens

Elas foram criadas de forma a balancear com eficiência todas as dimensões da rede utilizando um *compound coefficient*, sendo assim capaz de alcançar maior precisão e eficiência

Com base nesta estratégia de escalonamento das redes foram criadas oito versões desta arquitetura intituladas EfficientNetB0 (arquitetura-base) a EfficientNetB7 (obtidas escalando a arquitetura-base)

3 EfficientNet

EfficientNet nos permite formar features a partir de imagens que podem ser passadas posteriormente para um classificador, permitindo assim ser usada como um feature extractor network

Podemos implementar esse modelo de classificação de imagem de maneira simples utilizando o Keras, onde temos a opção de carregar os pesos do modelo que já foram pré-treinados utilizando o ImageNet

Workflow

1. Training Data Analysis: Aqui, entenderemos com que tipo de dados estamos lidando
2. Spectrogram: Em seguida, construiremos uma função para obter o espectrograma dos dados nos certificando de os gerarmos no formato correto (3D)
3. DataGenerator: Para lidar com o grande volume de dados, construiremos uma função capaz de dividi-los em batches, os carregando na RAM quando necessário
4. Model: Com os dados prontos para serem passados para a rede, passaremos para a construção do modelo
5. Predicting: Finalmente, seremos capazes de prever a probabilidade de cada file pertencente ao test data conter ou não o sinal simulado de ondas gravitacionais