



UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

POSGRADO EN ASTROFÍSICA

ASTROFÍSICA DE CAMPOS Y PARTÍCULAS

**OPTIMIZACIÓN DE LOS ALGORITMOS DE HAWC PARA LA BÚSQUEDA DE GRB's
ASOCIADOS A ONDAS GRAVITACIONALES**

TESIS

**QUE PARA OPTAR POR EL GRADO DE:
MAESTRO EN CIENCIAS (ASTROFÍSICA)**

PRESENTA:

JESÚS ABEL ROSALES GUZMÁN

Director de Tesis

**DR. HERMES LEÓN VARGAS
INSTITUTO DE FÍSICA-UNAM**

Ciudad Universitaria, Cd. Mx. Julio 2019



Universidad Nacional
Autónoma de México



UNAM – Dirección General de Bibliotecas
Tesis Digitales
Restricciones de uso

DERECHOS RESERVADOS ©
PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

Todo el material contenido en esta tesis esta protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

El uso de imágenes, fragmentos de videos, y demás material que sea objeto de protección de los derechos de autor, será exclusivamente para fines educativos e informativos y deberá citar la fuente donde la obtuvo mencionando el autor o autores. Cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por el respectivo titular de los Derechos de Autor.

Agradecimientos

Agradecemos el apoyo de el proyecto CONACyT 254964 y el Fondo Sectoral de Investigación para la Educación.

Resumen

Los destellos de rayos gamma (GRBs) son las explosiones más violentas del Universo conocido (después del Big Bang). Pueden alcanzar energías del orden de 10^{54} ergs [1]. Este tipo de destellos se caracteriza por no ser repetitivo, lo que implica que debe liberarse una enorme cantidad de energía en un intervalo de tiempo muy corto (del orden de segundos). Se han estudiado los progenitores de este tipo de eventos con el fin de desentrañar la naturaleza de la emisión de tan alta energía.

El observatorio HAWC (por sus siglas en inglés High Altitude Water Cherenkov) es un detector de rayos gamma y rayos cósmicos de muy alta energía (100 GeV-100 TeV). Se encuentra ubicado a 4100 m sobre el nivel del mar, en las faldas del volcán Sierra Negra en el Pico de Orizaba en Puebla, México. El detector consta de 300 tanques de 7.3 m de diámetro por 4.5 m de alto con 200,000 litros de agua especialmente tratada en cuyo interior se encuentran anclados 4 fotomultiplicadores (PMTs). La configuración del detector permite escanear 2/3 partes del cielo cada día por lo que es posible detectar eventos transitorios como GRBs.

El fotón más energético proveniente de un GRB fue el del GRB 130427A [2], el cual tuvo una energía de 95 GeV y se detectó a un redshift de 0.34. Dado que la distribución de GRBs es isotrópica en el cielo, es de esperarse que existan otros GRBs capaces de emitir fotones con energías similares o incluso mayores a la del GRB 130427A¹. De ahí que mejorar la capacidad de detección de fotones de bajas energías con HAWC haría posible la detección de GRBs como el que detectó el satélite de altas energías *Fermi*.

Con el fin de optimizar la búsqueda de rayos gamma de bajas energías (en el rango de 100 GeV - 1 TeV) se entrenó una red neuronal utilizando simulaciones de fotones y protones en el rango de energía mencionado. Se utilizó la información relevante de los eventos simulados y se aplicó un análisis de componentes principales (ACP) para transformar el espacio original de 39 variables² a 9 variables o componentes principales.

La red neuronal usa como entrada las 9 variables obtenidas del ACP y obtiene una precisión en la clasificación de eventos del 76 %. Para validar el desempeño de la red neuronal se utilizaron datos de la Nebulosa del Cangrejo. El conjunto de datos de la Nebulosa del Cangrejo corresponde a una muestra de 189 horas en las que se encuentra presente dicha fuente. Se hicieron comparaciones de significancia utilizando dos tipos de cortes de selección: el corte oficial en bajas energías que no incluye la discriminación de cascadas hadrónicas/electromagnéticas y el corte de predicciones de la red neuronal. Esta comparación arrojó un aumento de significancia del 43 % de la red neuronal con respecto al corte oficial.

Con el resultado de mejora en significancia se procedió a buscar datos correspondientes al GRB 170817A y GW 170817 visto por LIGO (Laser Interferometer Gravitational-Wave Observatory)-Virgo, *Fermi*-GBM (Gamma-ray Burst Monitor) y otros telescopios a menores energías. Cabe mencionar que la colaboración de HAWC hizo un estudio del trigger no encontrando nada significativo en la búsqueda. Los datos analizados corresponden a 2 horas de datos que fueron tomados entre las 20:55:09 UTC³ y las 22:55:09 UTC del 17 de Agosto de 2017, tiempo en el que la región en la que se identificó la procedencia de la onda gravitacional (GW) por LIGO-Virgo estuvo en el campo de visión de HAWC.

Se realizó también un estudio del campo de visión de HAWC de los días 15 a 20 de Agosto 2017 entre las 19:30:00 y 21:30:00 UTC. Se encontró una región con significancia de 3.96σ y se buscó alguna asociación con fuentes del catálogo TeVCat y de Fermi encontrando una posible asociación con el blazar Mrk 421.

¹La notación para nombrar a un GRB es mediante la fecha de su detección en formato aammss año, mes y día

²Cada cascada es caracterizada por alrededor de 64 variables que incluyen el número de evento, el estado de la reconstrucción, entre otras cosas. Se consideraron 39 variables que podrían ayudar a clasificar entre cascadas iniciadas por hadrones y por fotones.

³Universal Time Coordinated (UTC) Es el principal estándar de tiempo

La búsqueda de una contraparte electromagnética en altas energía del evento GW 170817 utilizando este nuevo método arrojó resultados negativos. Sin embargo, la mejora en los algoritmos de HAWC permitió encontrar una posible asociación con una fuente de TevCat. Se espera que se pueda utilizar esta mejora de los algoritmos para aumentar la sensibilidad de HAWC para fotones en el rango de cientos de GeV, tanto en la búsqueda de nuevas fuentes con emisión de rayos gamma, como en la búsqueda de fuentes transitorias.

Índice general

Agradecimientos	2
Lista de Acrónimos	12
1. Destellos de Rayos Gamma (GRBs)	13
1.1. Historia	13
1.1.1. El modelo de la <i>bola de fuego</i>	15
1.2. Progenitores de GRBs cortos	17
1.2.1. Fusión de Estrellas de Neutrones	17
1.2.2. Emisión de <i>Kilonova</i>	19
1.3. Distribución en distancias de GRBs cortos	20
1.4. Detección	20
2. Ondas Gravitacionales	23
2.1. LIGO, el primer detector de ondas gravitacionales	23
2.2. Método de análisis	24
2.2.1. Coalescencia de objetos compactos	25
2.2.2. Destellos de OG	27
2.3. GW150914	27
2.4. GW170817 y GRB170817A	27
2.4.1. Emisión temprana del GRB 170817A	29
3. Rayos gamma de muy alta energía, el observatorio HAWC	33
3.1. Cascadas atmosféricas	33
3.2. Reconstrucción de cascadas atmosféricas	34
3.2.1. Simulación	34
3.2.2. Calibración	35
3.2.3. Selección de Hits y definición de Bines	35
3.2.4. Reconstrucción del núcleo	35
3.2.5. Reconstrucción de la dirección	37
3.3. Separación Gamma/Hadrón	38
3.4. Estimación de la Energía	39
3.4.1. <i>Ground Parameter</i>	39
3.4.2. Neural Network	40
3.5. Variables y tipos de archivos	41
3.6. Búsquedas de GRBs	42
4. Creación de Mapas	45
4.1. LiFF	45
4.1.1. Formalismo de verosimilitud	45
4.1.2. Cálculo de la verosimilitud	46
4.2. Respuesta del detector	47
4.3. Mapas de significancia	47
5. Análisis de Componentes principales	49
5.1. Desviación estándar, varianza y covarianza	49
5.1.1. Matriz de covarianza y de correlación	50
5.2. ACP, el método	50

6. Deep Learning: Aprendizaje con redes neuronales	53
6.1. Inteligencia Artificial	53
6.2. <i>Machine Learning</i>	53
6.3. Aprendiendo representaciones de los datos	54
6.4. Lo diferente de <i>Deep Learning</i>	54
6.5. Red Neuronal en acción	56
7. Análisis y resultados	61
7.1. Preparación de la simulación	61
7.1.1. Componentes principales	61
7.1.2. Entrenamiento de la red	62
7.2. Comprobación de la red	64
7.2.1. El <i>Crab strip</i>	65
7.3. Búsqueda del GRB 170817A	66
8. Conclusiones y trabajo a futuro	71
A. Códigos desarrollados para el análisis	73
Referencias	83

Índice de figuras

1.1.	Distribución de T_{90} para 427 GRBs de BATSE	14
1.2.	Espectro del GRB 990123	15
1.3.	Distribución angular de eventos de BATSE	16
1.4.	Esquema del modelo de <i>bola de fuego</i> para GRBs	17
1.5.	Catálogo de GRBs de Fermi al 14-02-2018	18
1.6.	Escenario 1 de fusión de NSs	19
1.7.	Escenario 2 de fusión de NSs	20
1.8.	Distribución de corrimientos al rojo de GRBs	21
2.1.	Deformaciones ocasionadas por una GW	24
2.2.	Esquema de un detector como el de LIGO	25
2.3.	Etapas de la fusión de objetos compactos	26
2.4.	Evento GW 150914	28
2.5.	Ubicación de las señales de GWs , de rayos gamma y ópticas	30
2.6.	Seguimiento del evento de GWs realizado por diversos observatorios	31
3.1.	Fotografía del Observatorio HAWC	33
3.2.	Vista lateral de 3 tipos de cascadas simuladas	34
3.3.	Ajustes de la distribución de energía	36
3.4.	Evento de rayos gamma de la Nebulosa del Cangrejo visto desde HAWC	37
3.5.	Función de distribución lateral	37
3.6.	Comparación de una cascada electromagnética y una hadrónica desde el punto de vista de HAWC	38
3.7.	Distribución lateral de carga una cascada hadrónica y una electromagnética	39
3.8.	Ajuste de la función lateral	40
4.1.	Mapas de cuentas de todo el cielo	47
5.1.	ACP para reducir la dimensión del sistema original	51
5.2.	Un <i>Scree Plot</i> que sirve para identificar el número de componentes principales adecuado	51
5.3.	Componentes principales para un espacio multidimensional	52
6.1.	Ejemplo de red neuronal para clasificación de dígitos	55
6.2.	Representaciones del modelo de red neuronal	55
6.3.	Objetivo de una red neuronal	56
6.4.	Función <i>relu</i>	58
6.5.	Función <i>sigmoid</i>	58
6.6.	Gráficas de pérdida y precisión en entrenamiento y validación	59
7.1.	Concordancia entre la simulación y datos reales para un bin del análisis de HAWC	62
7.2.	<i>Scree Plot</i>	64
7.3.	Pérdida y precisión en entrenamiento y validación con respecto a la época (iteración sobre el conjunto de datos). El sobreajuste comienza a partir de la época 40, donde la pérdida y la precisión en validación comienzan a empeorar con respecto al entrenamiento.	65
7.4.	Pérdida y precisión para entrenamiento con sobreajuste	66
7.5.	Pérdida y precisión para la configuración final de la red	67
7.6.	Histograma de predicciones	68
7.7.	Significancia de la Nebulosa del Cangrejo con respecto al valor de $rec.NNPred$	68
7.8.	Mapas de significancia de la Nebulosa del Cangrejo	69
7.9.	Mapas de significancia en la ubicación del GRB 170817A	69

7.10. Mapa del campo de visión de HAWC con las fuentes de TevCat	70
7.11. Mapa de significancias alrededor de Mrk421	70
7.12. Mapa del campo de visión de HAWC con las fuentes del catálogo 3FGL de Fermi	70

Lista de Acrónimos

- ACP** Análisis de Componentes Principales
- AGILE** Italian Astro-rivelatore Gamma a Immagini LEggero
- BAT** Burst Alert Telescope
- BATSE** Burst And Transient Source Experiment
- BH** black hole
- CGRO** Compton Gamma Ray Observatory
- CORSIKA** COsmic Ray SIMulations for KASCADE
- DL** Deep Learning
- EBL** Extragalactic Background Light
- FAR** False Alarm Rate
- FITS** Flexible Image Transport System
- GBM** Gamma-ray Burst Monitor
- GCN** Gamma-ray Coordinates Network
- GeV** Gigaelectronvolt
- GLAST** Gamma-Ray Large Area Space Telescope
- GP** Ground Parameter
- GRB** Gamma Ray Burst
- GW** gravitational wave
- HAWC** High Altitude Water Cherenkov
- HEALPix** Hierarchical Equal Area isoLatitude Pixelation
- IA** Inteligencia Artificial
- IACT** Imaging Air Cherenkov Telescope
- INTEGRAL** International Gamma-Ray Astrophysics Laboratory
- KeV** Kiloelectronvolt
- LAT** Large Area Telescope
- LiFF** Likelihood Fitting Framework
- LIGO** Laser Interferometer Gravitational-Wave Observatory

MeV Megaelectronvolt

ML Machine Learning

Mpc Megaparsec

NKG Nishimura-Kamata-Greisen

NN Neural Network

NS neutron star

PE Photoelectron

PINC Parameter for the Identification of Nuclear Cosmic rays

PMT Photomultiplier tube

PSF Point Spread Function

SFCF Super Fast Core Fit

SNR Signal to Noise Ratio

SWEETS Software for Weighting Events and Eventlike Things and Stuff

TDC Time to Digital Converter

TeV Teraelectronvolt

ToT Time over Threshold

UTC Universal Time Coordinated

WCD Water Cherenkov Detector

WD white dwarf

XCDF The eXplicitly Compacted Data Format

Capítulo 1

Destellos de Rayos Gamma (GRBs)

Los destellos de rayos gamma (GRBs por sus siglas en inglés Gamma Ray Burst) son las explosiones más energéticas del Universo. Son destellos cortos ($t \sim s$), muy intensos ($E \sim 10^{54}$ erg). Pueden encontrarse con energías principalmente en el rango de Megaelectronvolt (MeV) [3] y tienen una gran variedad de espectros y de duraciones. Los GRBs fueron descubiertos por primera vez por los satélites Vela mediante observaciones entre Julio de 1969 y Julio de 1973 donde se detectaron 16 [4] y desde entonces han sido ampliamente estudiados.

En la sección 1.1 se da una introducción sobre la historia general de los GRBs y los avances que se hicieron principalmente en el siglo pasado. Después de esta introducción general, se habla sobre los GRBs cortos, distribución en distancias, progenitores y detección.

1.1. Historia

La detección en los años 60 s del primer GRB dio lugar a la aparición de diversas preguntas: ¿Cuáles son sus progenitores? ¿Acaso son estrellas normales o estrellas compactas como estrellas de neutrones (del inglés neutron star y de ahora en adelante **NS**) enanas blancas (white dwarf, de ahora en adelante **WD**) o agujeros negros (black hole, de ahora en adelante **BH**)? ¿Acaso los GRB ocurren en lugares del medio interestelar? ¿Qué tan intensos son? ¿Ocurren en nuestra galaxia o son objetos extragalácticos?. Resolver todas estas preguntas involucraba diversos problemas observacionales, como la localización precisas de estos GRBs en el cielo. Esta fue una tarea muy difícil en la banda de los rayos gamma donde las direcciones podrían ser estimadas de manera poco precisa. Una posible solución a este problema sería buscar a los GRBs en bandas de longitud de onda mayores. A pesar del gran esfuerzo que se hizo con satélites, hubo pocos resultados sobre la localización de GRBs.

El esfuerzo observacional más grande se llevó a cabo con el experimento BATSE (del inglés Burst And Transient Source Experiment) a bordo del Compton Gamma Ray Observatory (CGRO) [5]. CGRO fue lanzado en Abril de 1991 y fue sacado de órbita en 2000. En esos 9 años BATSE detectó 2702 GRBs [6] y gracias a este experimento se observaron muchas propiedades de la emisión temprana de los GRBs:

- La bimodalidad en las duraciones de los GRBs. Con BATSE se encontró que los GRBs muestran una distribución bimodal en cuanto a las duraciones, con dos picos anchos centrados cerca de 0.3 s y 20 s con un mínimo alrededor de 2s. Esta distribución bimodal permitió separar a los GRBs en dos grandes categorías, destellos cortos ($T_{90} < 2s$)¹ y destellos largos ($T_{90} > 2s$) [7] figura 1.1.
- Estructura temporal fina. Gracias a la gran área efectiva de BATSE se encontró que algunos GRBs pueden presentar estructuras en forma de pequeños picos con duraciones menores a milisegundos [8].
- Distribución espectral de la radiación. Usando los resultados de BATSE, Band y sus colaboradores [9], encontraron que la mejor descripción del espectro promediado de un GRB, desde 20 KeV hasta algunos MeV era una ley de potencias a pedazos a la que se le conoce ahora como **Función de Band**

¹ T_{90} se define como el tiempo para el cual se tiene el 90% de la emisión

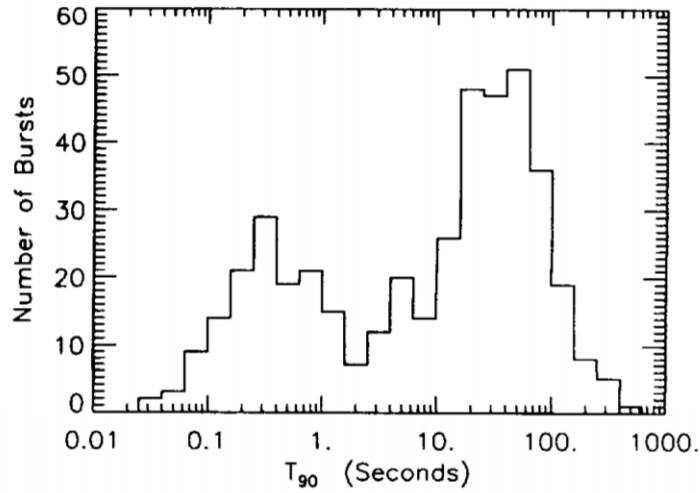


Figura 1.1: Distribución del T_{90} para 427 destellos observados con BATSE. Imagen tomada de [7].

(ver panel superior de la figura 1.2):

$$N(E) = A \begin{cases} \left(\frac{E}{100\text{keV}}\right)^\alpha \exp(-E/E_0) & \text{para } (\alpha - \beta)E_0 \geq E; \\ \left[\frac{(\alpha - \beta)E_0}{100\text{keV}}\right]^{(\alpha - \beta)} \exp(\beta - \alpha) \left(\frac{E}{100\text{keV}}\right)^\beta & \text{para } (\beta - \alpha)E_0 \leq E \end{cases} \quad (1.1)$$

donde α y β son los índices de la ley de potencias para bajas energías (por abajo de E_0) y altas energías (por arriba de E_0) respectivamente, A es el parámetro de normalización. Es sabido que los valores de estos parámetros cambian de un GRB a otro, con valores típicos de $\alpha = -1$, $\beta = -2.3$ y $E_0 = 150\text{KeV}$. Con esta descripción el espectro de la función de Band es no térmico. Si se grafica $E^2 N(E)$ vs E (la potencia por cada década de energía) puede verse que el espectro es no térmico. La nueva función denotada también como el espectro νF_ν (donde F_ν es el espectro de energía y ν es la frecuencia) generalmente muestra un máximo en la energía de los fotones $E_p = E_0(2 + \alpha)$ si $\beta - 2$ [1]. Esto es lo que pasa por lo general en los GRBs (ver panel inferior de la figura 1.2).

- Distribución angular de los GRBs en el cielo. A pesar de la dificultad de BATSE para ubicar con precisión a los GRBs en el cielo, si fue capaz de resolver un problema muy esperado: su distribución angular, es decir, los GRBs se distribuyen de manera isotrópica en el cielo[11]. La figura 1.3 muestra la distribución angular obtenida con BATSE para 2704 eventos. Con este resultado se descarta la hipótesis de que los GRBs se originan en el disco de nuestra Galaxia.
- Distribución de intensidades

Otros descubrimientos importantes sobre GRBs

El primer *afterglow*. Después de los descubrimientos de BATSE, el satélite italiano BeppoSAX reporta por primera vez el descubrimiento de un *afterglow* en rayos X correspondiente al GRB 970228 [12]. En este trabajo se encontró que el destello visto en rayos X contenía una fracción significativa de la energía total del GRB. La rápida localización de este evento permitió iniciar una campaña de búsqueda en múltiples longitudes de onda que culminó con la identificación de un evento transiente en el óptico [13] consistente con el evento de rayos X reportado por BeppoSAX.

El primer *redshift*. Otro evento importante en la historia de los GRBs ocurrió en Mayo del mismo año 1997, el satélite BeppoSAX reporta el primer *redshift* de un GRB. El GRB 970508 con $0.83 < z < 2.3$ [14]. A esta distancia y asumiendo una emisión de energía isotrópica se pudo establecer que $E_{iso} = (0.61 \pm 0.13) \times 10^{52}$ erg.

La conexión Supernova/GRB. Otro gran avance en el estudio de GRBs fue el descubrimiento de un GRB asociado a una Supernova. El GRB ocurrió el 25 de abril de 1998 (GRB 980425) y en su ubicación

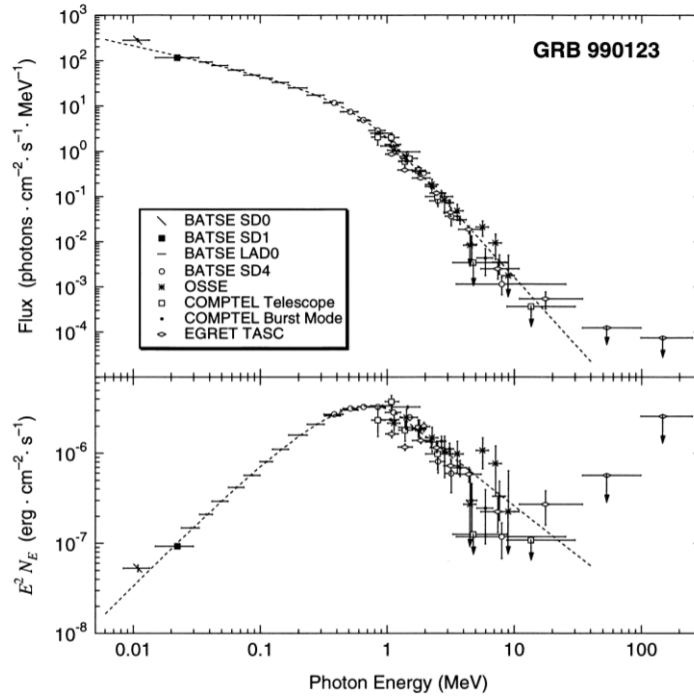


Figura 1.2: Ejemplo de un espectro de fotones provenientes de un GRB y su correspondiente función $E^2N(E)$. Se muestra el espectro del GRB ocurrido el 23 de enero de 1999 (GRB 990123), el cual fue detectado y localizado por el satélite BeppoSAX. Imagen tomada de [10].

se detectó una supernova (SN1998bw) en el radio y el óptico [15]. Este evento se detectó a un redshift de $z=0.0085$ (37 Mpc) con una energía isotrópica $E_{iso} = (8.5 \pm 0.1) \times 10^{47}$ erg.

La relación $E_{peak} - E_{iso}$. Dada la lejanía de los GRBs y la transparencia del Universo a la emisión en rayos gamma y rayos X, los GRBs podrían ser una herramienta muy poderosa para estudiar las regiones del Universo que son invisibles a la emisión en óptico. Dado que la medición del redshift de los GRBs se hacía basándose en observaciones de espectros en óptico e infrarrojo, la posibilidad de usar a los GRBs como faros del Universo oscuro descansa en medir sus luminosidades intrínsecas utilizando solamente rayos X y rayos gamma. Esto es equivalente a realizar una búsqueda de candelas estándar. Actualmente un GRB no puede ser considerado una candela estándar (como las a conocidas supernovas tipo Ia). Establecer una conexión entre la luminosidad de un GRB y algún otro parámetro detectable sólo en rayos gamma podría ser un avance importante. En este sentido BeppoSAX reportó la detección de una fuerte correlación entre la energía $E_{p,i}$ a la cual el espectro intrínseco νF_ν de los GRBs tiene un pico y la energía total liberada en el destello con la hipótesis de una emisión isotrópica. Esta relación se conoce como *relación de Amati* [16] y ha sido confirmada por misiones subsiguientes.

1.1.1. El modelo de la *bola de fuego*

Dada la escala de distancia cosmológica a la que se encontraban los GRBs, se requería que éstos liberaran una enorme cantidad de energía (hasta $\sim 10^{55}$ erg como el caso del GRB 080916C) en un intervalo de tiempo muy pequeño (decenas de segundos), además de tener un espectro no térmico y variabilidad corta (por abajo de la escala de tiempo de ms). Esta variabilidad corta, Δt , implica que se debe producir una enorme cantidad de energía en un volumen muy pequeño ($R < c\Delta t$), es decir, que se forme una *bola de fuego*, un concepto ya introducido en la época de los '70s por Cavallo y Rees [17]. Sin embargo, si se considera una bola de fuego estacionaria, debido a la alta sección eficaz de interacción $\gamma + \gamma \rightarrow e^+ + e^-$, los fotones de alta energía (> 1 MeV) serían suprimidos hasta ser degradados por abajo del límite de producción de pares, pero estos fotones si se observan. Además dada la alta opacidad de la *bola de fuego* estacionaria, se esperaría una emisión de radiación térmica y lo que en realidad se mide es radiación no térmica [1]. De ahí que el escenario preferido para un GRB sea el de una *bola de fuego* que se expande de manera relativista (escenario propuesto para GRBs en los '80 s [18]). Ahora, si la región de emisión se está expandiendo de manera relativista con un factor de Lorentz Γ , los fotones observados estarán corridos al azul por un factor $(1 + \Gamma)$ [1], la escala dimensional R correspondiente a la variación temporal Δt se vuelve $R < \Gamma^2 c\Delta t$, la opacidad a electrones asociados a bariones decrece como Γ^4 y el límite para producción de pares incrementa como Γ (véase por

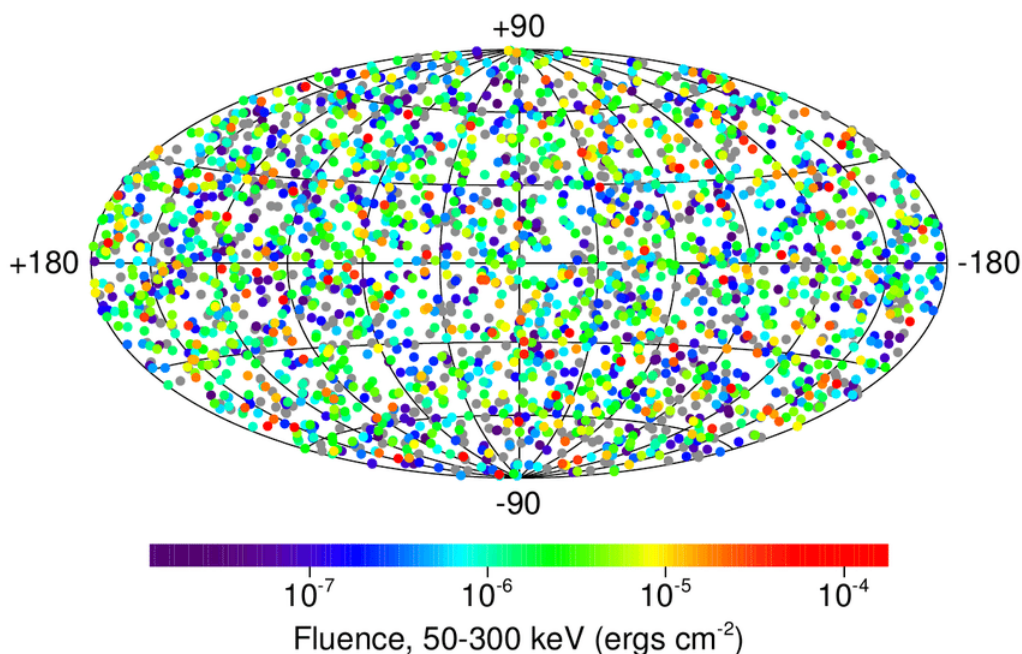


Figura 1.3: Distribución angular de 2704 eventos detectados por BATSE en coordenadas Galácticas. La *fluencia* se define como la energía total (en el tiempo) emitida por el objeto astrofísico en cuestión. Imagen tomada de https://apod.nasa.gov/apod/image/0006/grbmap_batse.gif.

ejemplo [19]).

En este escenario de *bola de fuego* con expansión relativista, se espera que la *bola de fuego* inicie con un radio R_{in} y energía E_0 , con masa bariónica M_0 (carga bariónica) mucho menor que E_0/c^2 . Dada la enorme profundidad óptica, la expansión radial de la bola debe ser resultado de una luminosidad súper Eddington en la cual la energía interna se está convirtiendo rápidamente en energía cinética[1]. Toda esta energía puede convertirse a radiación a través de choques, ya sea entre capas internas de la misma bola de fuego (choques internos) o con el medio externo (choques externos) (veanse por ejemplo [20, 21]). Como consecuencia de estos choques el factor de Lorentz decrece con el tiempo. Además, en presencia de campos magnéticos turbulentos (~ 100 G) que se pudieron haber formado detrás de los choques, los electrones deberían producir radiación de sincrotrón [22]. Se espera que la radiación de sincrotrón pierda energía conforme la *bola de fuego* en expansión se va frenando. De esta forma inicialmente se espera que la radiación se emita en rayos gamma (durante la fase temprana del GRB) y progresivamente evolucione a un *afterglow* en rayos X y después a uno en UV, óptico y radio [1]. Una representación del modelo de *bola de fuego* se encuentra en la figura 1.4. Este modelo tuvo un éxito inmediato dada su capacidad para explicar las propiedades espectrales y temporales de los *afterglow* de GRBs (véase por ejemplo [23]) por lo que se convirtió en el modelo estándar de GRBs.

Como puede verse en la figura anterior, el modelo de *bola de fuego* se centra en las regiones de emisión externas al motor central, es decir, a la fuente de todo este flujo relativista que le da poder al GRB. Este problema sobre el motor central se aborda tomando en cuenta las propiedades observadas del GRB. En particular los rangos energéticos, la escala de variabilidad y duración de un GRB sugieren que toda esa energía podría provenir de un disco de acreción masivo (~ 0.1 masas solares) alrededor de un objeto compacto, como podría ser un agujero negro [24].

Acerca de los progenitores del disco de acreción más un agujero negro existen diversos modelos: *merger* de NS-BH, colapsares o *merger* de WD-NS.

Nuevas misiones para búsqueda de GRBs

Los avances aportados por BATSE y BeppoSAX a la ciencia de GRBs permitieron la aprobación de nuevas misiones. La primera misión, lanzada en Noviembre de 2004 por la NASA se conoce como *Swift* (opera entre 0.2-10 KeV). La siguiente es GLAST (del inglés Gamma-Ray Large Area Space Telescope) lanzada en Junio de 2008 y que después del lanzamiento fue renombrada a *Fermi* (8 KeV hasta más de 300

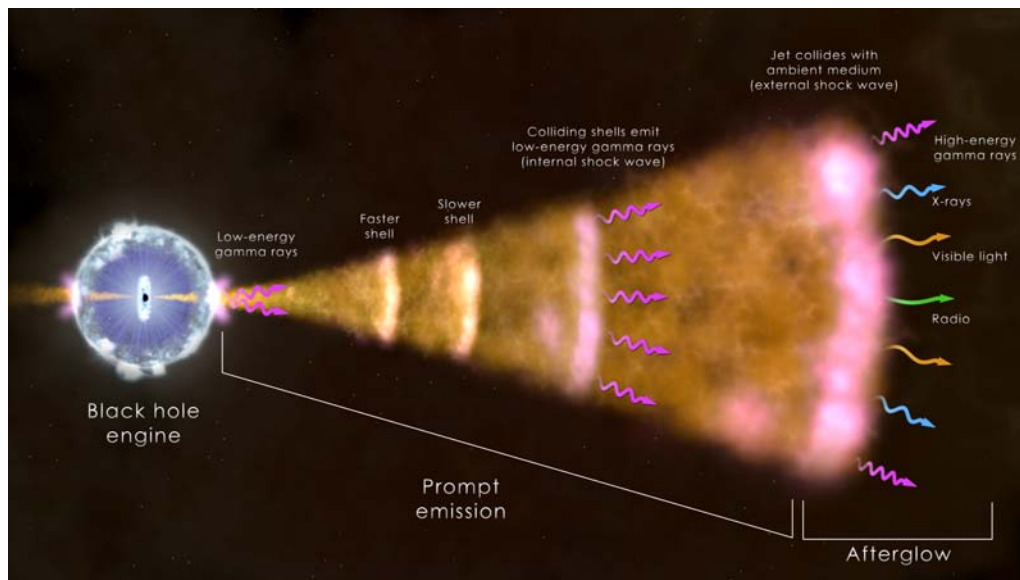


Figura 1.4: Esquema actualizado del modelo de *bola de fuego*, conocido también como el modelo estándar para GRBs. En esta imagen se asume como posible progenitor a una estrella masiva. Imagen tomada de <https://www.nasa.gov/content/goddard/nasa-sees-watershed-cosmic-blast-in-unique-detail>.

GeV). Otras misiones como INTEGRAL (del inglés International Gamma-Ray Astrophysics Laboratory) que abarca entre 15 KeV-10 MeV y AGILE (the Italian Astro-rivelatore Gamma a Immagini LEggero, 30 MeV-50 GeV y 10-40 KeV). La figura 1.5 muestra el catálogo de GRBs (hasta el 18 de Febrero de 2014) detectados con todos los instrumentos de *Fermi*.

GBM detecta alrededor de 240 GRBs por año, de los cuales 40-45 son GRBs cortos. Aproximadamente la mitad de las alertas de GBM se encuentran en el campo de visión de LAT (Large Area Telescope); de las cuales sólo el 10% son detectados por dicho instrumento, la mayoría son los GRBs más brillantes (tanto largos como cortos) que detecta GBM.

El telescopio *Swift*-BAT (Burst Alert Telescope) tiene menos área de cobertura que GBM pero a su vez es más sensible por lo que los destellos que son detectados por GBM y que quedan en el área de cobertura de BAT usualmente son detectados por BAT.

GBM y *Swift*-BAT tienen más de 30 detecciones en común cada año. El telescopio de rayos X (XRT) de *Swift* hace búsquedas de *afterglow* de los GRBs detectados por *Fermi*-LAT con el fin de permitir seguimientos en óptico y, en muchos casos, la determinación del *redshift*.

1.2. Progenitores de GRBs cortos

Con una rápida liberación de enormes cantidades de energía, la fusión de objetos compactos en un sistema binario era considerado uno de los modelos más aceptados como progenitor de GRBs en general. Esto con el conocimiento del primer pulsar binario (PSR 1913+16) descubierto por Hulse y Taylor en 1975 [26] además de las predicciones sobre la etapa final de sistemas binarios de **NS** con separación orbital pequeña; cuando las estrellas de neutrones radian en forma de ondas gravitacionales (**gravitational wave (GW)**) tienden a aproximarse, lo que inevitablemente lleva a que se fusionen. Dicho sistema binario puede estar conformado por **NS** o por **BH** y **NS**. En el caso de sistemas **NS-NS**, la fusión puede dejar como remanente un agujero negro rodeado de un disco de polvo que está siendo acretado. En el caso de **NS-BH** puede darse el mismo remanente si la estrella de neutrones es destruida por mareas fuera del horizonte del agujero negro [3].

1.2.1. Fusión de Estrellas de Neutrones

A continuación se describen brevemente los dos escenarios posibles de fusión de **NS** con producción de GRBs en la etapa final. En ambos escenarios el sistema está conformado por dos estrellas de secuencia

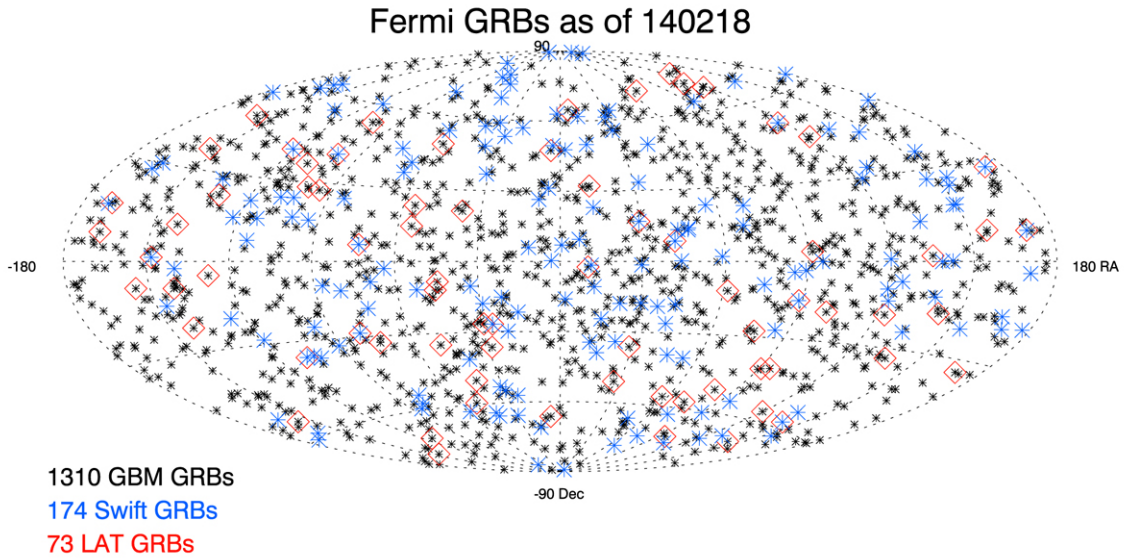


Figura 1.5: El catálogo de GRBs detectados por Fermi hasta el 18 de Febrero de 2014. Imagen tomada de [25].

principal (Main Sequence, MS), con una separación orbital $\lesssim 10$ UA (Unidades Astronómicas). La estrella principal tiene una masa M_P y la secundaria tiene una masa M_S

Escenario 1

La figura 1.6 ilustra el escenario que se describe a continuación

- La estrella primaria, de mayor masa, evoluciona fuera de la secuencia principal y se expande, llenando su lóbulo de Roche y transfiriendo masa a la estrella secundaria.
- La primaria entonces explota como supernova y deja una estrella de neutrones.
- Se tiene un sistema binario de **NS** con una estrella masiva.
- Este sistema pasará a través de una fase de binaria de rayos X, la cual evoluciona a fase de envoltente común conforme la estrella secundaria se expande.
- Durante esta fase de envoltente común la **NS** orbita en forma de espiral a la estrella masiva y la energía liberada expulsa la cubierta de hidrógeno. Se forma entonces un sistema de **NS** y estrella de He.
- Una vez que la estrella de He explota como supernova queda un sistema de **NS**, que con el tiempo se fusionarán formando un GRB acompañado de una onda gravitacional (del inglés gravitational wave, de ahora en adelante **GW**).

Escenario 2

La figura 1.7 ilustra el proceso que se describe a continuación.

- La secundaria evoluciona fuera de la secuencia principal antes de que la primaria explote como supernova.
- Las dos estrellas entran a una fase de envoltente común que consiste en dos núcleos de He orbitando entre sí dentro de una envoltente combinada de H (la envoltente está formada por el H proveniente de ambas estrellas).
- Cuando la envoltente de H es eyectada, las estrellas de He explotan.
- Si las explosiones de supernova no rompen la binaria, entonces se puede formar un sistema de **NSs**.
- El sistema puede de esta forma evolucionar para formar un GRB.

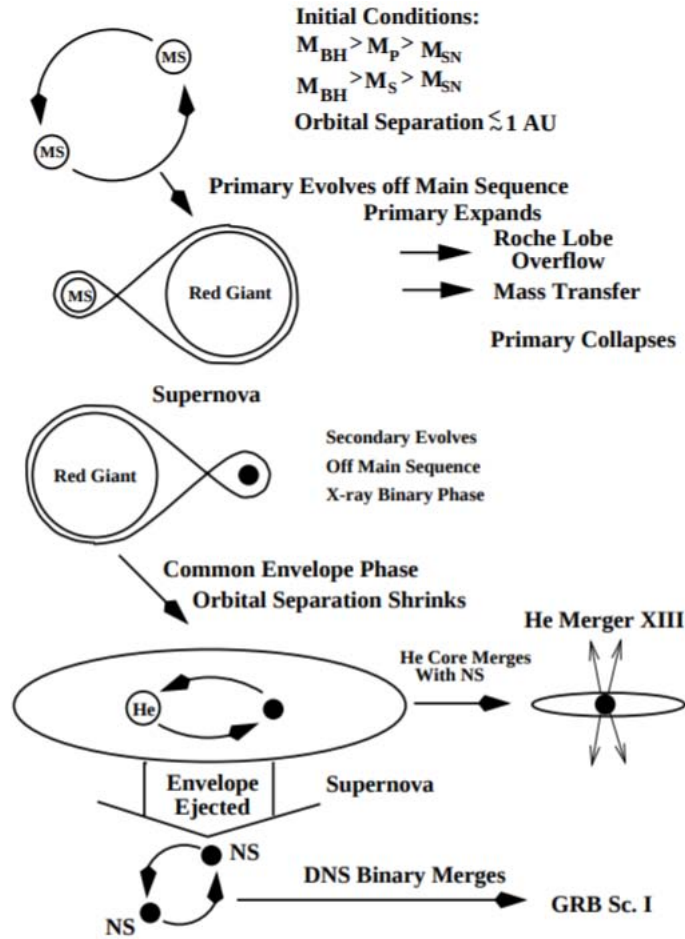


Figura 1.6: Escenario de producción de GRBs a partir de la fusión de NSs. M_{BH} , M_P y M_S son la masa del BH, la masa de la estrella primaria y la masa de la estrella secundaria respectivamente. $M_{SN} \sim 10M_{\odot}$ es la masa crítica por arriba de la cual las estrellas colapsan para formar supernovas. Las estrellas en secuencia principal se encuentran representadas con *MS* (del inglés main sequence) mientras que las NS se encuentran representadas como *NS*. El escenario XIII de formación de un GRB ocurre si la NS se fusiona con la estrella de He durante la fase de envoltorio común. Imagen tomada de [27].

La emisión de GW puede ser detectada con los detectores LIGO y Virgo a distancias de 40 Mpc para este tipo de fusión [28].

1.2.2. Emisión de *Kilonova*

Una de las predicciones importantes del modelo de fusión de objetos compactos es la eyección de materia rica en neutrones, ya sea de manera dinámica durante la fusión o mediante vientos provenientes del disco de acreción [29, 30]. Mediante simulaciones de fusión de objetos compactos (veanse por ejemplo [31, 32]) se han inferido los rangos de masas y velocidades de dichas eyecciones, los cuales están entre $M_{ey} \sim 10^{-3} - \sim 10^{-2}M_{\odot}$ y $v_{ey} \sim 0.1 - 0.3c$. Además la masa del material eyectado incrementa conforme mayor es la asimetría de masa entre los objetos compactos (es decir, la cantidad de material eyectado es mayor para fusión de NS-BH que NS-NS) (ver por ejemplo [33]). La energía cinética de dicha eyección puede alcanzar entre $E_k \sim 10^{49} - 10^{51}$ erg y al interactuar con el medio ambiente puede producirse emisión en radio. La escala de tiempo y el máximo de brillo estarán determinados por el medio ambiente, la velocidad de la eyección y la energía cinética [3].

La formación de elementos radiactivos pesados se produce mediante el *proceso-r*. El decaimiento radiactivo de estos elementos produce una fuente de calentamiento. Dada la rápida expansión del material eyectado, éste se vuelve ópticamente delgado muy rápidamente y con una luminosidad menor que una explosión de supernova. Además de las propiedades del material eyectado, la escala de tiempo, luminosidad y distribución espectral de energía, la kilonova también depende de la opacidad del material resultante del proceso-r ya que el pico en la curva de luz se espera cuando la opacidad se vuelve suficientemente baja para

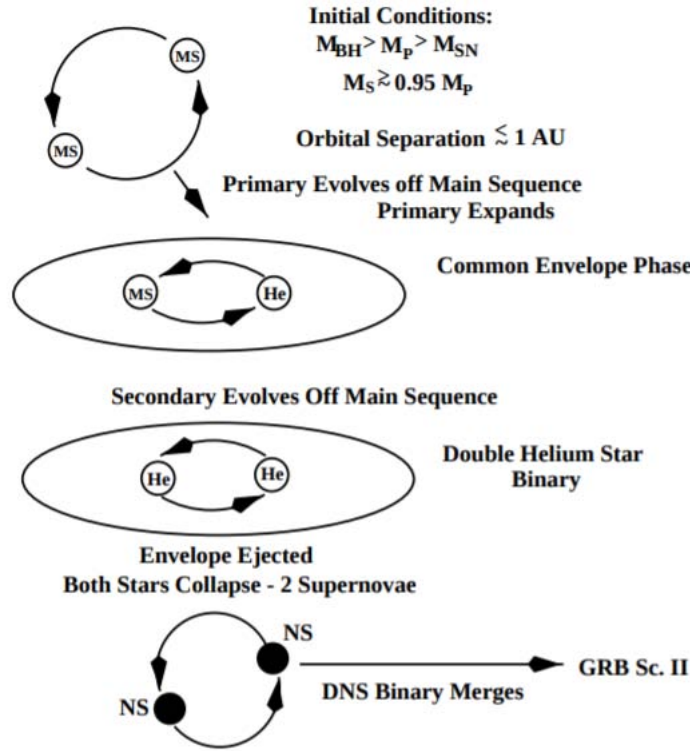


Figura 1.7: Escenario 2 de producción de GRBs a partir de la fusión de NSs. M_{BH} y M_P son las masas del BH y la estrella primaria respectivamente. M_{SN} se encuentra descrita en la figura 1.6. Las NS se encuentran representadas como NS y las de secuencia principal con MS . Imagen tomada de [27].

que los fotones puedan difundirse en el tiempo de escala de expansión, el cuál está dado como:

$$t_d = \frac{B\kappa M_{ej}}{cR}, \quad (1.2)$$

donde $B \approx 0.07$ es un factor geométrico, κ es la opacidad y $R = vt$ es el radio del material eyectado en expansión [3].

1.3. Distribución en distancias de GRBs cortos

En general es difícil obtener mediciones del corrimiento al rojo de los GRBs y usualmente esto se hace mediante espectroscopia de sus galaxias anfitrionas [34]², de ahí que la asociación con galaxias anfitrionas sea de interés particular para los estudios de GRBs. Es sabido que la asociación de un GRB con la galaxia anfitriona sólo puede asegurarse si se ha detectado el afterglow mezclado con la luz de la galaxia dentro de un rango de precisión menor a segundos de arco (arcsec) [34].

El grueso de los corrimientos al rojo de GRB cortos medidos se encuentra entre $z \sim 0.1 - 1.3$ lo que corresponde a distancias mayores a 420 Mpc, además es probable que las galaxias anfitrionas más débiles, para las cuales no hay mediciones exactas de corrimiento al rojo, se encuentren a $z \gtrsim 1$ [36]. La figura 1.8, adaptada de [3], muestra la distribución en corrimientos al rojo tanto de GRBs cortos como de GRBs largos; en promedio, los GRB cortos se encuentran a $\langle z \rangle \approx 0.48$ y si se agregan aquellas galaxias más débiles éste promedio incrementa a $\langle z \rangle \approx 0.63$.

1.4. Detección

¿Por qué es tan importante detectar la emisión en altas energías de los GRB?. Observar un corte espectral en la emisión de alta energía de un GRB puede traer consigo gran cantidad de información, ya sea de la

²De hecho el GRB 130603B es de los pocos GRBs con un corrimiento al rojo (z) bien calculado por medio de espectroscopia de absorción de su afterglow. Esto fue posible debido a que es un GRB muy brillante, con un $T_{90} = 0.18$ s, un espectro duro y un retraso espectral despreciable. Además pudo ser clasificado sin duda alguna como un GRB corto (véase por ejemplo [35])

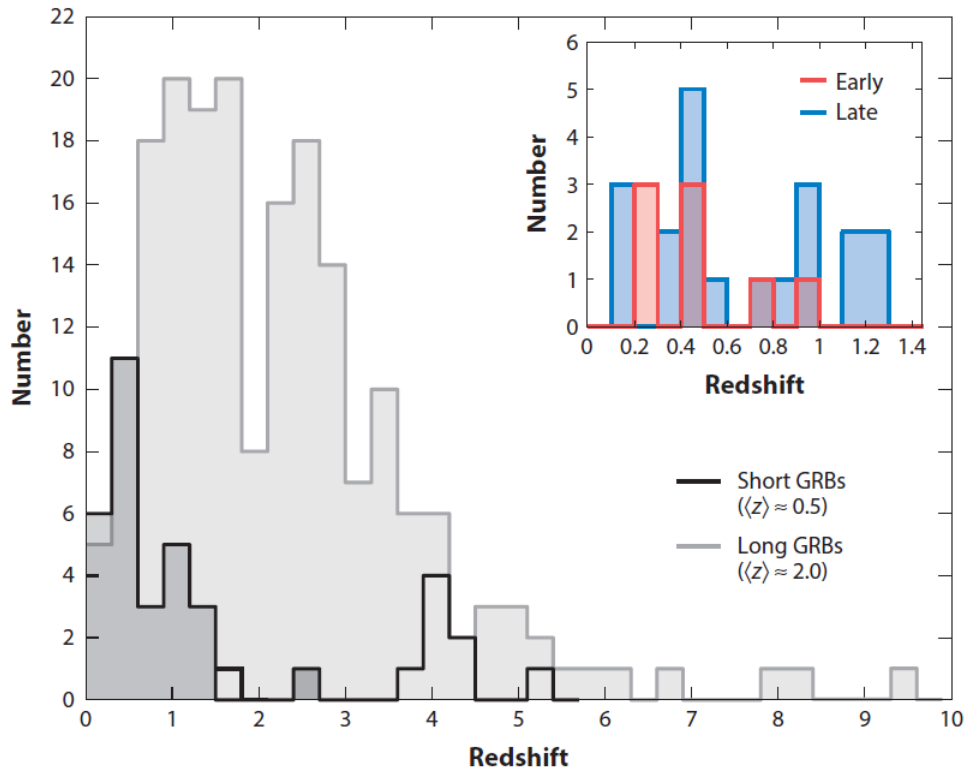


Figura 1.8: Distribución de corrimiento al rojo de GRBs cortos y largos. El histograma interno es la distribución de corrimiento al rojo de GRBs cortos como función de la galaxia anfitriona, en este caso no hay una clara diferencia entre galaxias tempranas (galaxias elípticas) o tardías (galaxias espirales) como anfitrionas.

fuelle en sí o de la propagación de los fotones gamma a través del medio interestelar (ISM). Como ejemplo, es posible conocer el factor de Lorentz (Γ) y con ello restringir modelos de jet. Cuando los rayos gamma se propagan a través del medio interestelar sufren atenuación debida a interacciones con la luz de fondo extragaláctica (EBL por sus siglas en inglés)[37]. En consecuencia la detección de un corte en la distribución espectral de energía puede derivar en el calculo de la densidad del EBL.

Así como los ejemplos anteriores, existen diversos resultados que podrían obtenerse con la observación en altas energías de la emisión de GRBs. De ahí que existan tantas instalaciones centradas en la búsqueda de dicha emisión. Actualmente existen tres clases principales de detectores de altas energías: satélites [38, 39], IACTs³ [40] y arreglos de detectores de cascadas atmosféricas [41]. Los satélites poseen amplios campos de visión con un ciclo operacional cercano al 100% pero su reducido tamaño evita que puedan reunir suficiente estadística como para llegar a energías mayores a decenas de GeV. Por otro lado los IACT, que operan en un rango arriba de ~ 50 GeV tienen gran sensibilidad, y gran resolución angular y en energía, sin embargo este tipo de detectores sólo funcionan con buenas condiciones climáticas y noches despejadas sin luna y su campo de visión está restringido a 5 grados o menos en diámetro. Finalmente los detectores de cascadas atmosféricas, como HAWC tienen un campo de visión más grande ($\sim 2sr$ o 16% de $4sr$) y un ciclo de funcionamiento cercano al 100%.

En el capítulo 3 se hablará sobre los resultados del observatorio HAWC en la búsqueda de GRBs

³De Imaging Air Cherenkov Telescope, que son observatorios que detectan luz Cherenkov en el aire como el telescopio Magic.

Capítulo 2

Ondas Gravitacionales

Una de las diferencias fundamentales entre la teoría de la relatividad general de Einstein ([84], [85]) y la teoría de Newton de la gravedad es la predicción de Ondas Gravitacionales: oscilaciones de la métrica espacio-temporal que se propagan a la velocidad de la luz. Este fenómeno se mantuvo como una predicción de la teoría durante más de 50 años hasta la primera evidencia observacional de su existencia con el descubrimiento y posterior seguimiento del sistema binario PSR 1913+16 por Russell Hulse y Joseph Taylor [26]. Este sistema binario consiste de dos **NS** que orbitan la una a la otra con un periodo de 7.75 hrs. Los pulsos en radio provenientes de este sistema, monitoreados durante varios años, demostraron que el periodo orbital va decreciendo de acuerdo con la predicción relativista de emisión de **GW** [42]. En alrededor de 300 millones de años, la órbita de este sistema binario decrecerá de tal forma que las **NS** coalescerán para formar un objeto compacto en un proceso que producirá **GWs** detectables.

Esencialmente, la teoría de la relatividad general dice que la masa y la energía producen una curvatura del espacio-tiempo de cuatro dimensiones y que la materia se mueve como respuesta a esta curvatura. Las ecuaciones de campo de Einstein describen la interacción entre masa y curvatura del espacio tiempo, algo equivalente a lo que describen las ecuaciones de Maxwell sobre las interacciones entre cargas eléctricas y campos magnéticos. Así como las ondas electromagnéticas se producen como solución en el tiempo a las ecuaciones de Maxwell en el vacío, las **GWs** son las soluciones dependientes del tiempo a las ecuaciones de campo de Einstein en el vacío. Las **GWs** pueden pensarse como perturbaciones de naturaleza oscilatoria a una métrica del espacio-tiempo plana, o de Minkowsky, o de manera equivalente pueden pensarse como un esfuerzo oscilante al que están sujetas dos masas de prueba.

Las **GWs** viajan a la velocidad de la luz y son de carácter transversal, es decir, las *deformaciones* que provocan ocurren en direcciones ortogonales a la dirección de propagación de la onda. A diferencia de las ondas electromagnéticas, que son de naturaleza dipolar, las **GWs** son de naturaleza cuadrupolar: el patrón de deformación contrae el espacio a lo largo de una dirección mientras lo expande a lo largo de la dirección ortogonal en el plano transversal (figura 2.1).

Otra cosa que diferencia las **GWs** de las ondas electromagnéticas es que las primeras viajan por el espacio sin sufrir perturbaciones ya que interactúan muy débilmente con la materia. Además se considera que las **GWs** son intrínsecamente no lineales ya que la densidad de energía de la onda genera una curvatura adicional del espacio-tiempo [43]¹. Las **GWs** son tan débiles que uno no espera detectar ondas producidas en la Tierra; deben buscarse en objetos astrofísicos muy masivos moviéndose a velocidades relativistas. Como ejemplo fuentes de **GWs** que puedan existir en nuestra galaxia o en otras galaxias podrían producir deformaciones en la Tierra que no sobrepasen una parte en 10^{21} .

2.1. LIGO, el primer detector de ondas gravitacionales

Como ya se mencionaba, se requiere de un interferómetro de Michelson para medir los patrones de deformación producidos por el paso de una **GWs**. LIGO (del inglés Laser Interferometer Gravitational-Wave Observatory) utiliza tres interferómetros de Michelson: En Hanford se encuentran dos, H1 y H2 con una longitud de sus brazos de 4 km y 2 km respectivamente y en Livingston se encuentra un tercero, L1, con una longitud de 4 km. Los brazos de cada interferómetro se encuentran colocados de manera ortogonal

¹Cabe mencionar que este efecto sólo es significante cuando se está muy cerca de grandes fuentes de ondas

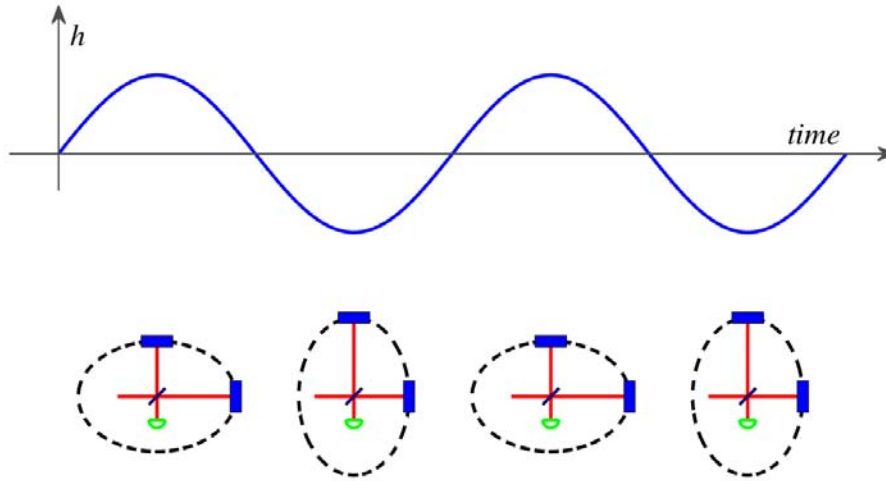


Figura 2.1: Una **GW** que pasa perpendicular al plano de la figura se caracteriza por la amplitud de deformación h . La onda distorsiona un anillo de partículas de prueba para formar una elipse, elongada en una dirección en medio ciclo de la onda y en el siguiente ciclo elongando la elipse en la otra dirección. Esta distorsión puede medirse con un interferómetro de Michelson orientado como en la figura. Las oscilaciones en la longitud de los brazos del interferómetro modulan los cambios de fase de la luz, lo cual se observa como modulaciones de intensidad en los fotodetectores (semicírculos verdes). Imagen tomada de [43].

formando una L (figura 2.2). Estos tres interferómetros son esencialmente los mismos excepto por la longitud de H2.

En la figura 2.2 se muestra un esquema del detector LIGO, el cual consta de dos brazos de longitud 4 km cada uno a través de los cuales se propaga un láser. Cuando una **GW** pasa por el detector de manera ortogonal al plano de éste alargará uno de los brazos que lo componen y acortará el otro durante medio ciclo de la onda. El fotodetector registra estas variaciones diferenciales en las cavidades de los brazos. El ruido de los detectores se muestra en el recuadro b). Como puede verse, la mayor sensibilidad del detector se encuentra entre 100-300 Hz, que es donde el ruido debido a fuentes no relacionadas con **GWs** es más bajo. Cuando una onda gravitacional pasa por el detector se espera que haya un cambio de longitud en los brazos: midiendo el tamaño de dicha deformación es posible encontrar la amplitud de la onda gravitacional ecuación (2.1).

$$\Delta L(t) = \delta L_x - \delta L_y = h(t)L \quad (2.1)$$

Donde h ($\sim 10^{-20}$) es la amplitud de la onda gravitacional que produce la deformación. Esta variación de longitud afecta directamente la diferencia de fase entre los láseres que regresan al divisor de haz, transmitiendo una señal óptica proporcional a la deformación producida en el detector.

Con el uso de detectores como el de la figura 2.2 colocados en puntos muy alejados entre sí, es posible distinguir entre ruido ambiental local y ondas gravitacionales. La astronomía de ondas gravitacionales se vale de este método para dar un estimado de la ubicación en el cielo y la polarización de la onda gravitacional. Aunque es cierto que con LIGO pueden hacerse estudios de **GWs**, utilizando una red de detectores a lo largo del planeta es posible mejorar el potencial astrofísico de su estudio. Durante todos estos años, la red de detectores se ha ido incrementando. El proyecto japonés TAMA (Japanese Interferometric Gravitational-Wave Project) construyó un interferómetro de 300 m a las afueras de Tokyo, el proyecto Británico-Alemania GEO (German-English Observatory; British-German Cooperation for Gravity Wave Experiment) construyó un interferómetro de 600 m a las afueras de Hanover, Alemania y el Observatorio Gravitacional Europeo construyeron el interferómetro de 3 km, *Virgo*, cerca de Pisa, Italia. La colaboración de LIGO realizó un acuerdo con Virgo para que todos los datos recolectados sean analizados de manera conjunta.

2.2. Método de análisis

LIGO fue diseñado para detectar **GWs** producidas por diferentes fuentes astrofísicas. Se considera que tales fuentes pueden ser de naturaleza continua o transitoria. El análisis de este tipo de señales depende de si han podido simularse las señales de onda o no. Las búsquedas de **GWs** se organizan en 4 grandes

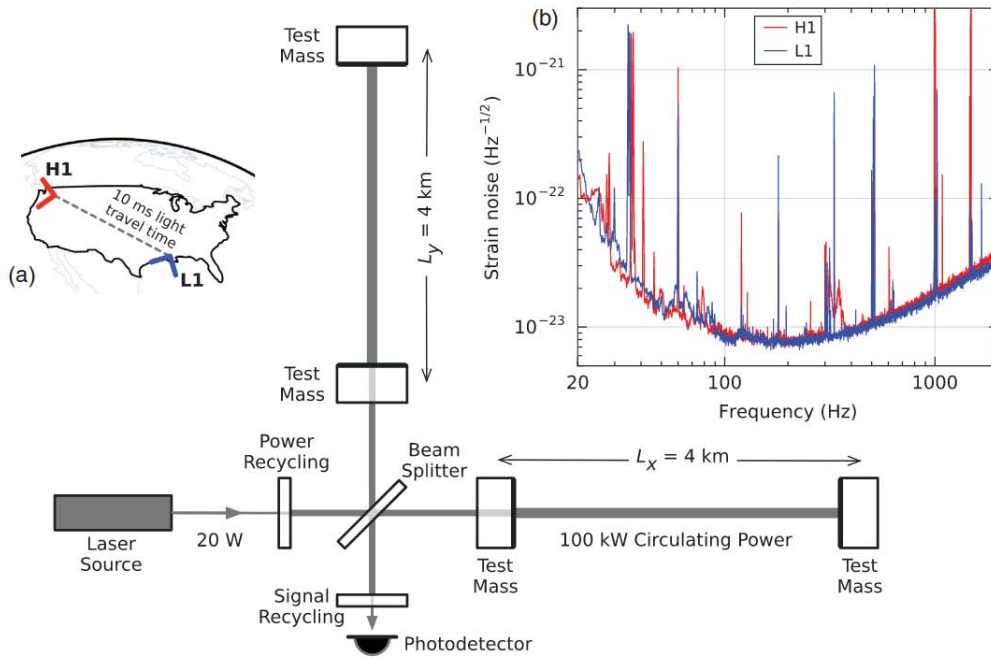


Figura 2.2: Esquema simplificado de un detector como el de LIGO. Una onda gravitacional propagándose de manera ortogonal al plano del detector y polarizada linealmente de manera paralela a uno de los brazos alargará uno de los brazos y acortará el otro durante medio ciclo de la onda; los cambios de longitud se invierten en el siguiente medio ciclo de la onda. El fotodetector de salida registra estas variaciones diferenciales de las cavidades de los brazos. En este caso la respuesta direccional es máxima, pero sigue siendo significativa aún para diferentes direcciones de llegada y polarizaciones de la onda gravitacional. Recuadro a) Localización de los dos detectores que conforman LIGO: H1 se encuentra en Hanford y L1 en Livingston. Recuadro b) El ruido de ambos detectores para tiempos cercanos a la detección de la señal. Imagen tomada de [44].

categorías, de las cuales sólo se describirá el método de análisis de las primeras dos ya que son el tipo de señal correspondiente a GW 150914 y GW 170817-GRB 170817.

- *Formas de onda transitorias modeladas: búsqueda de coalescencia de sistemas binarios de objetos compactos.* Las etapas finales en el *inspiral* de sistemas binarios son las fuentes transitorias de **GWs** mejor entendidas y modeladas. Cada componente de estos sistemas binarios puede ser **NS** o **BH**. Para este tipo de sistemas se puede modelar una forma de onda con gran precisión.
- *Formas de onda transitorias no modeladas: la búsqueda de destellos de **GWs**.* Dentro de esta categoría caen supernovas, *mergers* de **BHs** y sismos de **NS**, los cuales no pueden ser modelados totalmente. En estos casos se requiere un análisis más general.
- *Formas de onda de banda angosta: la búsqueda de fuentes continuas de **GWs**.* Un ejemplo dentro de esta categoría sería una **NS** rotante que no está perfectamente alineada con su eje de rotación.
- *Formas de onda de banda ancha: búsqueda de un fondo de **GWs**.* Algún tipo de proceso existente en el Universo temprano podría haber producido un fondo continuo pero estocástico de **GWs**.

2.2.1. Coalescencia de objetos compactos

La coalescencia de objetos compactos en un sistema binario proporcionaría evidencia irrefutable sobre la existencia de **BHs**. Además, múltiples observaciones de este tipo proporcionarían información sobre la población de estos sistemas en el Universo hasta una distancia de cientos de Mpc [43]. La coalescencia de sistemas que involucren **NS** proveerían información acerca de la ecuación de estado en tales condiciones extremas. Como ya se vio en el capítulo anterior, se piensa en estos objetos como progenitores de los GRB cortos.

La forma de onda del *inspiral* de un sistema binario de objetos compactos es un *chirp*: una onda sinusoidal

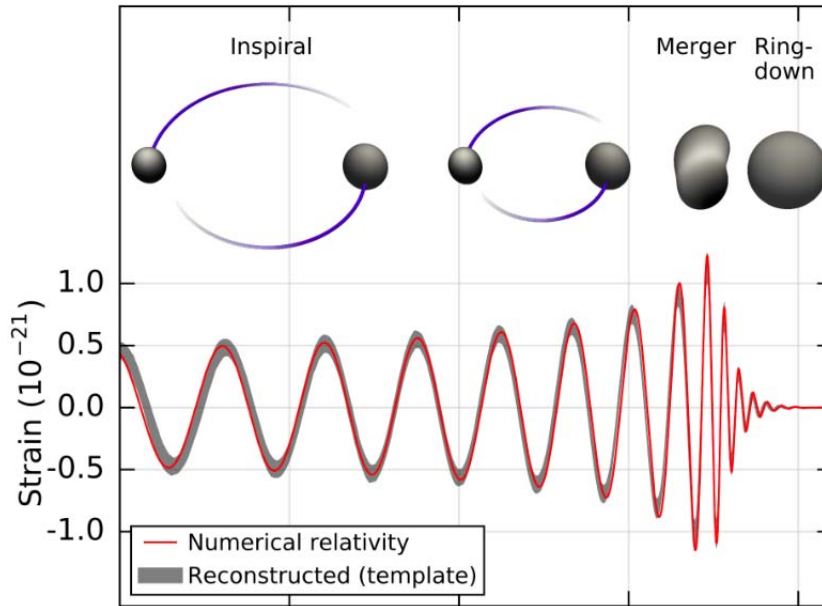


Figura 2.3: Etapas de la fusión de objetos compactos (en este caso dos BHs). El *inspiral* ocurre mientras los objetos compactos se aproximan entre sí, el *merger* es la etapa en la cual los objetos compactos se fusionan y el *ringdown* ocurre cuando el BH recién formado sigue oscilando hasta quedar estático. Imagen tomada de <https://www.ligo.org/science/Publication-GW150914/>.

que incrementa en frecuencia y amplitud hasta el final de la fase de *inspiral*². La fase de *inspiral* de un sistema binario de **NS** (suponiendo una masa de $1.4M_{\odot}$ para cada **NS**) completaría cerca de 2000 vueltas en la banda de LIGO en un intervalo de tiempo de decenas de segundos antes del *merger*, emitiendo una **GW** con frecuencia máxima de alrededor de 1500 Hz [43]. La figura 2.3 ilustra las tres fases que ocurren durante la fusión de objetos compactos, a saber el *inspiral*, el *merger* y el *ringdown*.

Método de análisis

Ya que las formas de onda en el *inspiral* y el *ringdown* para cada par de masas (m_1, m_2) son bien conocidas, las búsquedas se basan en filtrar las señales usando un conjunto de plantillas correspondientes al espacio de parámetros de las masas (m_1, m_2). En la práctica para cubrir un intervalo de masas entre $2M_{\odot}$ y $35M_{\odot}$ se utilizan alrededor de 7000 plantillas.

Este proceso de filtrado de coincidencias genera un *trigger* al momento en que la tasa de señal a ruido (en inglés Signal to Noise Ratio (SNR)) en la salida del filtrado excede un valor límite. Este límite se establece mediante un balance entre dos factores: debe ser lo suficientemente bajo como para que se pueda hacer una buena estimación del fondo debido al ruido del detector y debe ser lo suficientemente alto para mantener una cantidad manejable de *triggers*. Cada *trigger* tiene asociada una plantilla en específico, es decir, un par de masas y un tiempo de coalescencia que maximiza la tasa de señal a ruido de ese evento.

Todos los *triggers* se generan de manera independiente en cada detector y la manera de disminuir la enorme cantidad es estableciendo un *trigger* de coincidencias entre ambos detectores, es decir, aquellos que corresponden a plantillas con masas y tiempos de coalescencia similares, dentro de ventanas de tiempo adecuadas.

Para evaluar la eficiencia de detección y la línea de análisis se inyectan una serie de eventos falsos. Estas formas de onda inyectadas cubren el rango más amplio posible del espacio de parámetros (masas de las componentes, espines, orientaciones, localizaciones en el cielo y distancias) [43].

²El *inspiral* ocurre cuando los dos objetos compactos se aproximan entre sí, el *merger* ocurre en el momento en que los objetos compactos se fusionan y el *ringdown* ocurre cuando el agujero negro recién formado sigue oscilando hasta quedar estático.

2.2.2. Destellos de OG

Además de las señales bien modeladas de las que se habló en la sección anterior, el observatorio LIGO busca *destellos* de **GWs**, que se definen como cualquier señal con una duración corta ($t \lesssim 1s$) y con una señal significativa en la banda de frecuencia de sensibilidad del detector ($45 \leq f \leq 2000$ Hz). El colapso del núcleo en una estrella masiva podría emitir **GWs** a través de distintos mecanismos [45]. Un *merger* de objetos compactos puede ser considerado como un destello de **GWs** sólo si la masa es tan grande que el grueso de la señal del *inspiral* queda por abajo del rango de sensibilidad del detector dejando sólo una pequeña parte del *merger* para ser detectada [43].

Como se verá más adelante, una gran cantidad de eventos de **GWs** podrán tener emisión de radiación electromagnética y/o partículas muy energéticas que serían detectadas por telescopios y otros instrumentos. Entonces, más allá de una búsqueda de **GWs** se estaría buscando una detección conjunta con la que se podría constreñir la ubicación y el tiempo del evento de **GWs**, permitiendo de esta forma una búsqueda más sensitiva.

2.3. GW150914

La primera detección de **GWs** ocurrió el 14 de septiembre de 2015 y corresponde a la fusión de un sistema binario de **BHs**. Los eventos detectados tenían una tasa de señal a ruido combinada de 24 [44]. La detección de los eventos de **GWs** entre uno y otro detector ocurre con una diferencia temporal de 10 ms. A dicha detección se le conoce como GW 150914 y no fue detectada más que por LIGO ya que Virgo se encontraba en mantenimiento y actualización mientras que GEO 600 no tuvo la sensibilidad suficiente para detectar el evento ya que no se encontraba en modo observador. Debido a que sólo dos detectores pudieron captar la **GW**, la única manera de saber la dirección de llegada es mediante la diferencia de tiempos de detección, con la cual se obtiene un área de alrededor de 600 deg^2 (región con credibilidad del 90 %) [46].

La señal coincidente de ambos detectores después de haber sido filtrada se muestra en la figura 2.4. Las características de GW 150914 apuntaron a la fusión de agujeros negros. Por arriba de 0.2 s la señal detectada aumenta en frecuencia y amplitud en alrededor de 8 ciclos desde 35 hasta 135 Hz, donde la amplitud llega al máximo. La manera más plausible de explicar este tipo de aumento de frecuencia y amplitud es mediante el *inspiral* de dos masas orbitando m_1 y m_2 debido a la emisión de OG. A frecuencias bajas, tal evolución está caracterizada por la masa del *chirp* [50]:

$$M = \frac{(m_1 m_2)^{3/5}}{(m_1 + m_2)^{1/5}} = \frac{c^3}{G} \left[\frac{5}{96} \pi^{-8/3} f^{-11/3} \dot{f} \right]^{3/5} \quad (2.2)$$

Donde f y \dot{f} son la frecuencia observada y su derivada, G y c son la constante gravitacional y la velocidad de la luz respectivamente. Para el evento GW 150914 se estimó una *chirp mass* $M \simeq 30M_\odot$.

La tabla 2.1 muestra todos los eventos de **GWs** registrados durante las diversas etapas de toma de datos de LIGO. Con excepción de GW 170817, todos los eventos corresponden a fusión de **BHs**.

2.4. GW170817 y GRB170817A

El 17 de Agosto del 2017 a las 12:41:04 UTC se detectó, a través de **GWs**, un candidato a coalescencia de **NS** (GW 170817). Este evento fue detectado por los detectores LIGO y Virgo. Además *Fermi* mediante GBM detectó de manera independiente un destello de rayos gamma (GRB 170817A). En esta sección se describirán algunos detalles de esta detección conjunta así como una descripción breve del seguimiento realizado en otras longitudes de onda.

La OG fue detectada como un *trigger* común, que fue diseminado mediante un GCN³. Un re-análisis de los datos de LIGO-Hanford, LIGO-Livingston y Virgo confirmaron una coincidencia significativa en la señal. Combinando los datos obtenidos de los tres detectores se logró el primer mapa del cielo a las 17:54:51 UTC. Este análisis inicial coloca a la fuente a una distancia de 40^{+8}_{-8} MPc en una región elongada de $\sim 31 \text{ deg}^2$ (90 % de credibilidad), centrada alrededor de una A.R. $\alpha(J2000.0) = 12^h 57^{min}$ y Dec $\delta(J2000.0) = -17^\circ 51'$ [51]. El siguiente mapa del cielo, producido mediante un segundo análisis entre la red de detectores y distribuido a las 23:54:40 UTC, fue consistente con el primero: una región del cielo de $\simeq 34 \text{ deg}^2$ (90 % de credibilidad) centrada alrededor de $\alpha(J2000.0) = 13^h 09^{min}$ y Dec $\delta(J2000.0) = -25^\circ 37'$.

³Gamma-ray Coordinates Network es el sistema encargado de distribuir las noticias de GRBs y otros eventos transitorios vistos por satélites

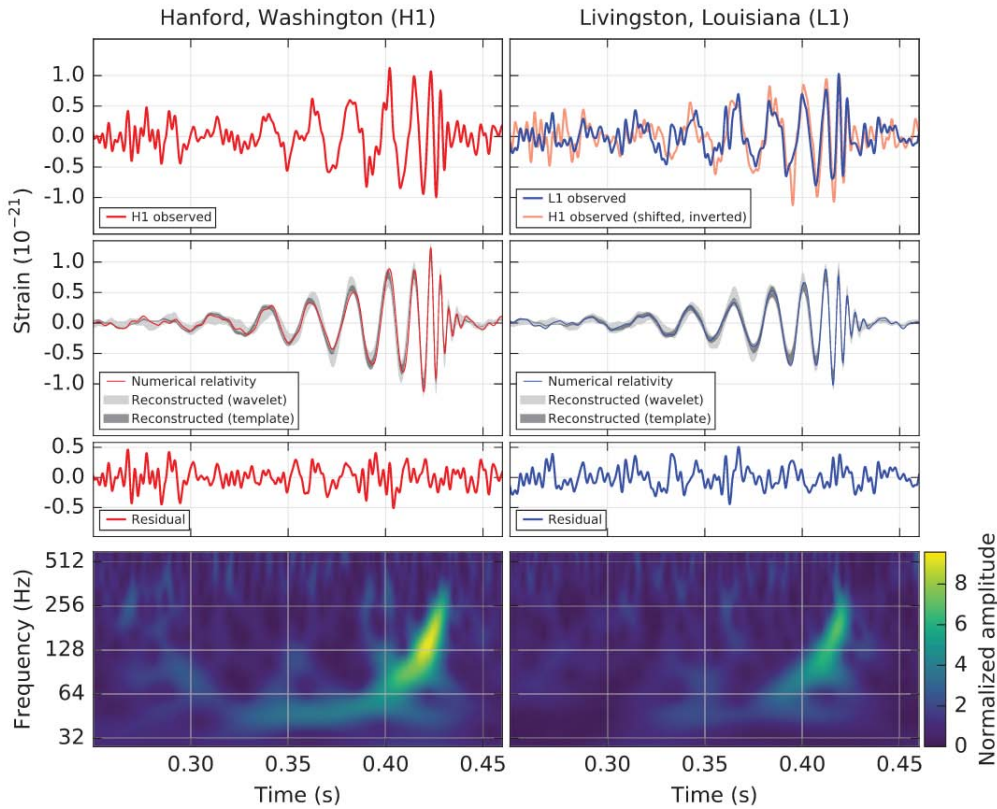


Figura 2.4: El evento de la onda gravitacional GW 150914 visto por el detector H1 (izq.) y L1 (der.). El tiempo mostrado es con relación al 14 de Septiembre de 2015 a las 09:50:45 UTC. La primera fila muestra la deformación en cada detector. En la segunda fila se ha aplicado un filtro de banda entre 35-350 Hz, las líneas sólidas muestran una forma de onda obtenida mediante relatividad numérica con los parámetros consistentes a los obtenidos a partir de GW 150914 [47]. Las partes sombreadas corresponden a áreas de 90 % de credibilidad para dos diferentes formas de onda reconstruidas; los modelos de señal hechos a partir de simulaciones de BHs binarios están marcados en gris oscuro mientras que en gris claro no se usaron modelos astrofísicos, en su lugar se utilizó una combinación lineal de paquetes de onda sinusoidales Gaussianos [48]. El residuo obtenido después de extraer las formas de onda de las series de tiempo del detector se observa en la tercera fila. La representación de tiempo-frecuencia de la deformación mostrando como aumenta la frecuencia de la señal en el tiempo se muestra en la parte baja de la figura [49]. Imagen tomada de [44].

El análisis *offline* de la onda gravitacional GW 170817 con los detectores LIGO-Hanford y LIGO-Livingston reportaron una tasa de falsa alarma (False Alarm Rate (FAR)) menor a 1 por cada 8.0×10^4 [52]. En cuestión de tiempos de detección, la **GW** se detectó primero en Virgo, 22 ms después a LIGO-Livingston y 3 ms después en LIGO-Hanford. Esta **GW** se detectó con una SNR combinada de 32.4^4 .

Los análisis de los modelos de formas de onda revelaron los siguientes valores: los rangos de masa de las componentes del sistema m_1 y m_2 son respectivamente $m_1 \in (1.36 - 2.26)M_\odot$ y $m_2 \in (0.86 - 1.36)M_\odot$. La masa de *chirp*, M , es el parámetro que domina la evolución en frecuencia de la radiación gravitacional durante la fase de *inspiral*. Ya que este parámetro domina la porción de la **GW** que se encuentra en la banda de sensibilidad de los instrumentos, fue el parámetro mejor medido, $M = 1.188^{+0.004}_{-0.002}M_\odot$ [51]. La masa total es $2.82^{+0.47}_{-0.09}M_\odot$ y la razón de las masas m_2/m_1 está ligada al rango 0.4-0.1. Estos resultados son coherentes con un sistema binario conformado por dos **NS** [51].

La figura 2.5 muestra la región donde se ubicó al evento GW 170817, GRB 170817A.

⁴La SNR del evento GW 150914 fue de 24 [44]

Evento	m_1/M_\odot	m_2/M_\odot	\mathcal{M}/M_\odot	M_f/M_\odot	$E_{rad}/(M_\odot c^2)$	d_L/Mpc	z	$\Delta\Omega/\text{deg}^2$
GW150914	$35.6^{+4.8}_{-3.0}$	$30.6^{+3.0}_{-4.4}$	$28.6^{+1.6}_{-1.5}$	$63.1^{+3.3}_{-3.0}$	$3.1^{+0.4}_{-0.4}$	430^{+150}_{-170}	$0.09^{+0.03}_{-0.03}$	179
GW151012	$23.3^{+14.0}_{-5.0}$	$13.6^{+4.1}_{-4.8}$	$15.2^{+2.0}_{-1.1}$	$35.7^{+9.9}_{-3.8}$	$1.5^{+0.5}_{-0.5}$	1060^{+540}_{-480}	$0.21^{+0.09}_{-0.09}$	1555
GW151226	$13.7^{+8.8}_{-3.2}$	$7.7^{+2.2}_{-2.6}$	$8.9^{+0.3}_{-0.3}$	$20.5^{+6.4}_{-1.5}$	$1.0^{+0.1}_{-0.2}$	440^{+180}_{-190}	$0.09^{+0.04}_{-0.04}$	1033
GW170104	$31.0^{+7.2}_{-5.6}$	$20.1^{+4.9}_{-4.5}$	$21.5^{+2.1}_{-1.7}$	$49.1^{+5.2}_{-3.9}$	$2.2^{+0.5}_{-0.5}$	960^{+430}_{-410}	$0.19^{+0.07}_{-0.08}$	924
GW170608	$10.9^{+5.3}_{-1.7}$	$7.6^{+1.3}_{-2.1}$	$7.9^{+0.2}_{-0.2}$	$17.8^{+3.2}_{-0.7}$	$0.9^{+0.0}_{-0.1}$	320^{+120}_{-110}	$0.07^{+0.02}_{-0.02}$	396
GW170729	$50.6^{+16.6}_{-10.2}$	$34.3^{+9.1}_{-10.1}$	$35.7^{+6.5}_{-4.7}$	$80.3^{+14.6}_{-10.2}$	$4.8^{+1.7}_{-1.7}$	2750^{+1350}_{-1320}	$0.48^{+0.19}_{-0.20}$	1033
GW170809	$35.2^{+8.3}_{-6.0}$	$23.8^{+5.2}_{-5.1}$	$25.0^{+2.1}_{-1.6}$	$56.4^{+5.2}_{-3.7}$	$2.7^{+0.6}_{-0.6}$	990^{+320}_{-380}	$0.20^{+0.05}_{-0.07}$	340
GW170814	$30.7^{+5.7}_{-3.0}$	$25.3^{+2.9}_{-4.1}$	$24.2^{+1.4}_{-1.1}$	$53.4^{+3.2}_{-2.4}$	$2.7^{+0.4}_{-0.3}$	580^{+160}_{-210}	$0.12^{+0.03}_{-0.04}$	87
GW170817	$1.46^{+0.12}_{-0.1}$	$1.27^{+0.09}_{-0.09}$	$1.186^{+0.001}_{-0.001}$	≤ 2.8	≥ 0.04	40^{+10}_{-10}	$0.01^{0.00}_{0.00}$	16
GW170818	$35.5^{+7.5}_{-4.7}$	$26.8^{+4.3}_{-5.2}$	$26.7^{+2.1}_{-1.7}$	$59.8^{+4.8}_{-3.8}$	$2.7^{+0.5}_{-0.5}$	1020^{+430}_{-360}	$0.20^{+0.07}_{-0.07}$	39
GW170823	$39.6^{+10.0}_{-6.6}$	$29.4^{+6.3}_{-7.1}$	$29.3^{+4.2}_{-3.2}$	$65.6^{+9.4}_{-6.6}$	$3.3^{+0.9}_{-0.8}$	1850^{+840}_{-840}	$0.34^{+0.13}_{-0.14}$	1651

Tabla 2.1: Parámetros estimados para las once detecciones reportadas en el catálogo de **GW** de LIGO y Virgo durante las corridas O1 y O2. Se reportan valores con 90% de nivel de credibilidad que incluyen errores estadísticos y sistemáticos provenientes de promediar los resultados de dos modelos de forma de onda para sistemas binarios de BHs. Las columnas mostradas son, de izquierda a derecha: masas de las componentes (m_i), *chirp mass* (\mathcal{M}), masa final del objeto (M_f), energía radiada (E_{rad}), distancia lumínica (d_L), corrimiento al rojo (z) y localización en el cielo ($\Delta\Omega$). Tomada de [93].

2.4.1. Emisión temprana del GRB 170817A

A las 12:41:20 UTC *Fermi*-GBM, a través de un GCN hace el primer anuncio del GRB 170817A. El GRB había ocurrido 14 s antes; a las 12:41:06 UTC. La localización final de *Fermi*-GBM acotó el GRB 170817A a una región del cielo en $\alpha(J2000.0) = 12^h 28^{min}$ y $\delta(J2000.0) = -30^\circ 37'$, la región de 90% de credibilidad cubría $\sim 1100 \text{deg}^2$ [53]. Posteriores seguimientos del trigger de *Fermi*-GBM determinaron la duración del destello: $T_{90} = 2.0 \pm 0.5 \text{s}$ con lo cual el GRB 170817A quedó clasificado como un GRB corto. Además se encontraron dos componentes del destello: un pulso inicial que va desde T0-0.320 s, con T0 el inicio del trigger, hasta T0+0.832s seguido por una cola menos intensa que va desde T0+0.832 s hasta T0+1.984 s. El espectro del pulso principal puede ajustarse con una ley de potencias con corte exponencial (función Comptonizada) con índice -0.62 ± 0.40 , $E_{peak} = 185 \pm 62 \text{KeV}$ y flujo promedio de $(3.1 \pm 0.7) \times 10^{-7} \text{ergcm}^{-2} \text{s}^{-1}$

Seguimientos posteriores

La figura 2.6 muestra los seguimientos hechos por diferentes observatorios durante los días siguientes a la detección del GRB y la **GW**. En particular para el observatorio HAWC, la región de localización dada por LIGO-Virgo se volvió visible el 17 de Agosto entre las 19:57 y las 23:25 UTC. En este caso no se realizó detección significativa pero se establecieron límites superiores para energías $> 40 \text{TeV}$ asumiendo un espectro de $E^{-2.5}$. A continuación se enlistan los seguimientos realizados en multi-longitudes de onda.

- En rayos gamma se hicieron seguimientos con Fermi, Integral, Swift, H.E.S.S., HAWC, AGILE entre otros. La detección en esta longitud de onda se dió casi inmediatamente después del evento de **GWs** (GRB 170817A).
- En rayos X se hicieron seguimientos con Swift, INTEGRAL, Chandra entre otros. El descubrimiento de la contraparte en rayos X de la **GW** se dió 9 días después.
- En UV los seguimientos se hicieron con Swift y con el telescopio espacial Hubble aproximadamente 1 día después del evento de la **GW**.
- Donde se llevó a cabo la mayor cantidad de seguimientos con gran número de telescopios fue en el óptico. Entre algunos de los telescopios se encuentra el telescopio espacial Hubble, Swope, Las Cumbres, Pan-STARRS1, ATLAS entre otros. La emisión en óptico de GW 170817 se dió a 10.86 horas del evento inicial.

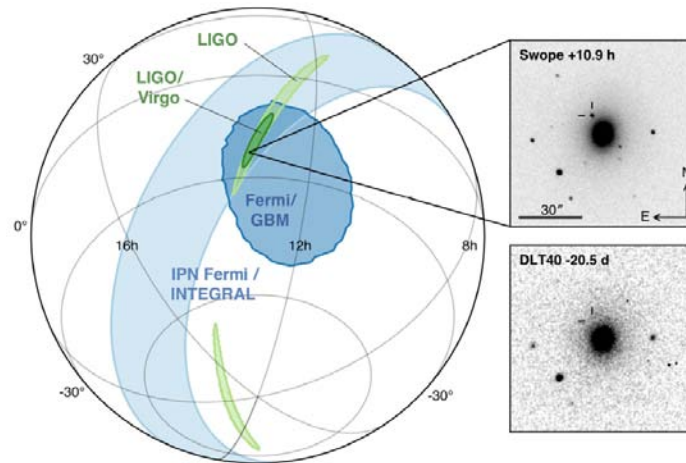


Figura 2.5: Localización de las señales en **GW**s, rayos gamma y ópticas. El panel izquierdo es una proyección de las regiones de 90 % de credibilidad de LIGO ($190deg^2$, verde claro), la localización inicial obtenida por LIGO y Virgo ($190deg^2$, verde oscuro), en azul claro la triangulación entre los tiempos de retraso de *Fermi* e INTEGRAL y *Fermi* con GBM (azul oscuro). Los recuadros muestran la aparente localización de la galaxia anfitriona NGC 4993 en una imagen óptica de *swope* tomada 10.9 hrs después del *merger* y una imagen del descubrimiento previo de DLT40 2.5 días antes del *merger*. Imagen tomada de [51].

- En infrarrojo se hicieron seguimientos con el telescopio espacial Hubble, Spitzer Gemini-South, VISTA entre otros. El descubrimiento de esta contraparte se dió aproximadamente 13 horas después de la **GW**.
- En radio los seguimientos se hicieron con el VLA, ASKAP, ALMA, VLBA, LOFAR, entre otros. El descubrimiento de esta contraparte se dió aproximadamente 10 días después del evento de la **GW**.

Podría decirse que este evento tuvo una de las campañas más grandes de búsqueda puesto que incluyó a una gran cantidad de satélites y telescopios. Todo este enorme esfuerzo se vió recompensado con las diversas detecciones del mismo evento en distintas longitudes de onda.

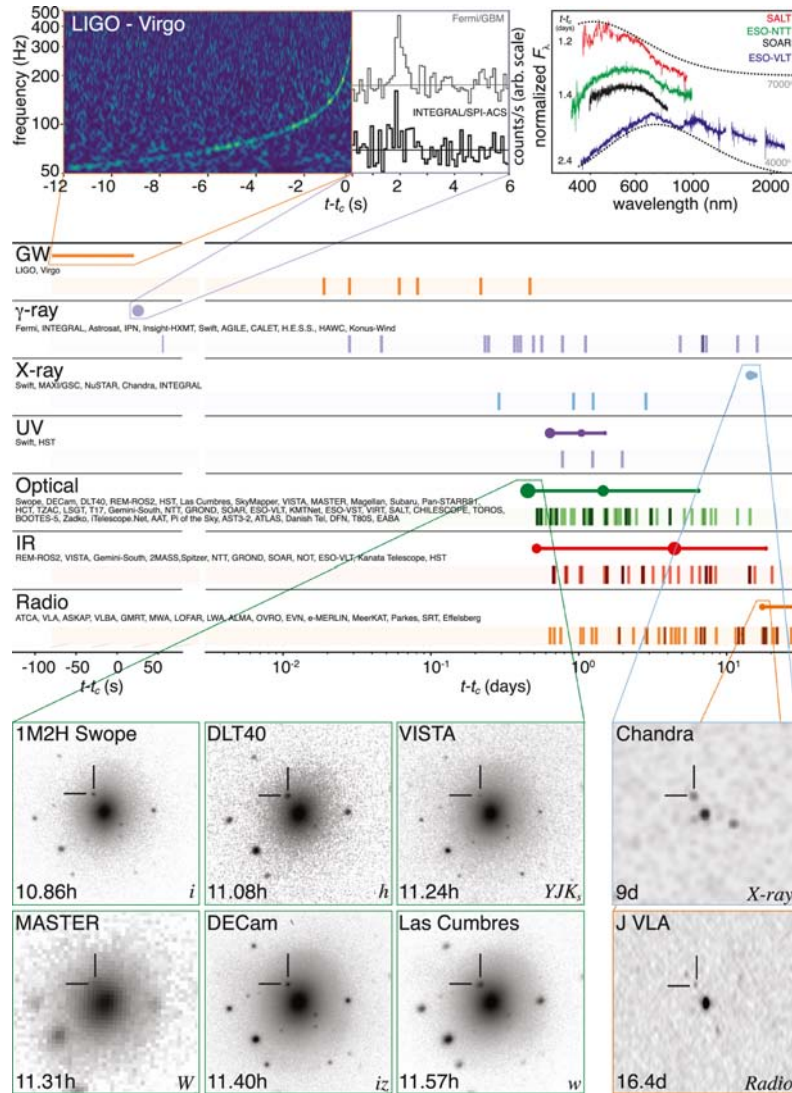


Figura 2.6: Seguimiento del evento GW 170817 y GRB 170817A. Todos los tiempos son con respecto al tiempo del *merger*. Se presentan dos tipos de información: las marcas verticales representan las veces que apareció información en un GCN. Las líneas horizontales sólidas indican el tiempo en el que el evento fue detectable en al menos uno de los telescopios que aparecen del lado izquierdo. Los puntos corresponden con observaciones.

Capítulo 3

Rayos gamma de muy alta energía, el observatorio HAWC

El observatorio HAWC, ubicado en las faldas del volcán Sierra Negra en el Parque Nacional Pico de Orizaba es un detector que utiliza el efecto Cherenkov en agua para estudiar los fotones gamma y los rayos cósmicos provenientes de lugares muy extremos del Universo. El detector está compuesto por 300 tanques (WCD o Water Cherenkov Detectors) de 7.5 m de diámetro por 5 m de alto (figura 3.1), 294 de los cuales están instrumentados. Cada tanque contiene alrededor de 200,000 litros de agua especialmente tratada y en el fondo están colocados 4 tubos fotomultiplicadores (PMT): 3 de 8" formando un triángulo equilátero de 3.2 m de lado y uno de 10" en el centro. Con esta configuración se escanean 2 terceras partes del cielo cada día con una sensibilidad a cascadas atmosféricas con energías que van desde 100 GeV hasta 100 TeV [54].



Figura 3.1: Fotografía del Observatorio HAWC tomada en el año 2016.

3.1. Cascadas atmosféricas

De acuerdo con la partícula primaria que inicia la cascada atmosférica, este tipo de eventos tienen dos nombres. Cuando la partícula primaria es un hadrón a la cascada producida se le conoce como **cascada hadrónica**. En cambio si la cascada fue originada por fotones o electrones, se le conoce como **cascada electromagnética**.

Las cascadas electromagnéticas son más compactas que las hadrónicas de energías comparables, es decir, el núcleo de una cascada electromagnética tiene mayor densidad de partículas que el núcleo de una cascada hadrónica. Esto se debe a que los electrones y fotones provenientes de la producción de pares y la radiación bremsstrahlung no adquieren el momento transversal característico de interacciones hadrónicas.

Las partículas electromagnéticas carecen de la cola de muones y hadrones tardíos. Además, las cascadas hadrónicas sufren grandes fluctuaciones durante su desarrollo, cosa que no ocurre con las cascadas electromagnéticas. Además, la componente muónica es muy baja en las cascadas electromagnéticas. Esto se debe a que los muones son resultado de interacciones fotonucleares, en particular de fotones-piones.

La figura 3.2 muestra la distribución lateral de tres tipos de cascadas.

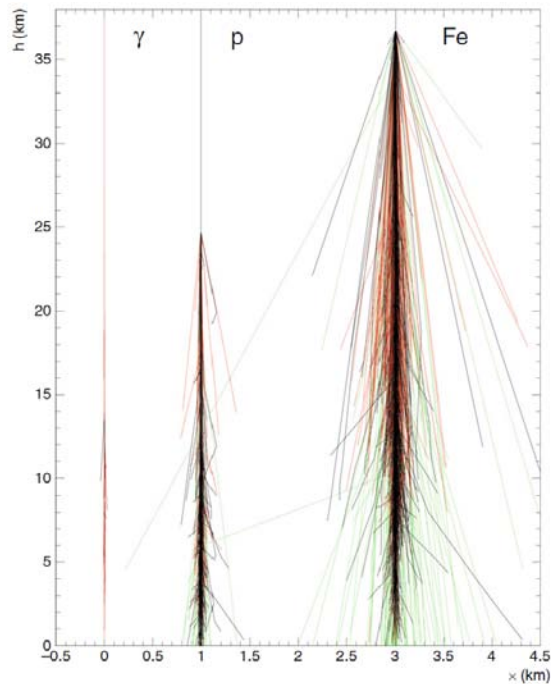


Figura 3.2: Vista lateral de 3 tipos de cascadas iniciadas por un fotón, un protón y un núcleo de Fe, cada partícula tiene energía 10^5 GeV. Las partículas que componen la cascada tienen energías > 10 GeV. En rojo se muestra la componente electromagnética, en negro la hadrónica y en verde los muones. Imagen tomada de [55].

3.2. Reconstrucción de cascadas atmosféricas

Una partícula primaria que atraviesa la atmósfera puede producir una cascada atmosférica. Las cascadas atmosféricas que atraviesan el detector dejan información a su paso. Parámetros como la carga depositada en los detectores (Water Cherenkov Detector (WCD)) o la distribución de tiempos de llegada permiten reconstruir la energía, la dirección de llegada y el tipo (rayo cósmico o rayo gamma) de la partícula que inició la cascada (partícula primaria) [56]. Es claro que existen diferentes fuentes de ruido que pueden activar el detector y que no son debidas a cascadas atmosféricas, por ejemplo pulsos tardíos de los PMT o ruido propio de estos mismos, entre otros. Para poder distinguir y guardar la información correcta se utiliza un *trigger* de multiplicidad el cual es básicamente la activación de un número determinado de PMTs (N_{thresh}) en una ventana de tiempo de 150 ns. El valor de N_{thresh} ha variado desde los inicios del observatorio y ha oscilado entre 20-50 PMTs; actualmente el valor usado es 28 PMTs. Los eventos que cumplen este *trigger* son almacenados en disco y procesados para su reconstrucción [57].

La reconstrucción de cascadas involucra determinar su dirección, la probabilidad de que el evento sea un fotón y el tamaño del evento. Como fondo se utilizan todas las cascadas reconstruidas e identificadas como hadrónicas y la forma de distinguirlas de las electromagnéticas es mediante su topología y la presencia de partículas muy penetrantes a nivel de tierra. Las siguientes subsecciones tratan el método para reconstrucción de cascadas atmosféricas.

3.2.1. Simulación

Dado que hacer simulaciones es una manera importante para conocer el comportamiento real del detector, se utiliza una combinación de paquetes libres de simulación y un software propio para simular HAWC. El paquete CORSIKA (COsmic Ray SIMulations for KAscade, <https://www.ikp.kit.edu/corsika/>) (V7.4000) se utiliza para simular cascadas atmosféricas y propagar partículas primarias desde la atmósfera hasta llegar a nivel de tierra. A partir de ahí se utiliza GEANT4 (<https://geant4.web.cern.ch/>) (v4.10.00) para propagar las partículas de la cascada a través del arreglo del detector y seguir los caminos de los fotones Cherenkov hasta llegar a los PMTs.

En la simulación también se agrega ruido de PEs que no está correlacionado. La eficiencia absoluta de detección de fotones Cherenkov se establece al escalar la respuesta de los PMTs simulados a muones verticales para igualar a los datos ¹. Para conocer la respuesta de los algoritmos de reconstrucción, los datos simulados son reconstruidos de la misma forma que los datos experimentales.

3.2.2. Calibración

El primer paso para la reconstrucción consiste en la calibración, que es un proceso mediante el cual se encuentra la relación entre la luz medida y la carga depositada. La calibración asocia el tiempo que pasa la señal de cada PMT por encima de un valor límite ² con el número de fotoelectrones. El sistema de calibración también toma en cuenta la respuesta de los PMTs a los cambios rápidos en la señal de entrada.

Ya que los subsecuentes algoritmos de reconstrucción tratan a todos los PMT como si fueran iguales, a pesar de que hay de 8 pulgadas y de 10 con diferentes eficiencias, es necesario hacer un escalamiento definiendo primero una carga efectiva Q_{eff} . Con esta definición los valores de los PEs del PMT central de 10 pulgadas se multiplican por 0.46 para tener la misma escala que los PMTs de 8 pulgadas [57].

Es necesario también tomar en cuenta la longitud de los cables que unen los PMT con la casa de conteo puesto que estos son de diferentes longitudes. Cada PMT tiene un desfase de tiempo que toma en cuenta las diferentes longitudes y algunos otros retrasos que pudieran ocurrir entre PMT y PMT. Los desfases se miden con precisión de nanosegundos y el sistema de calibración los refina hasta sub-nanosegundos mediante ajustes iterados de cascadas hadrónicas. Finalmente se aplica una rotación de $\sim 0.2^\circ$ [57] a todos los eventos para asegurarse de que la Nebulosa del Cangrejo se encuentre en la posición correcta ($83^\circ.63$ en A.R. y $22^\circ.01$ en Dec).

3.2.3. Selección de Hits y definición de Bines

El sistema de adquisición de datos (DAQ por sus siglas en inglés) guarda $1.5 \mu s$ de datos de todos los PMT que se activaron durante un evento de cascada atmosférica. Para hacer los ajustes de la cascada atmosférica se utiliza un subconjunto de datos de entre 150 ns previos y 400 ns posteriores al momento de activación del PMT. Para evitar considerar *afterpulses* se remueven aquellos *hits* que ocurren instantes después registrar *hits* con mucha carga. Finalmente cada canal (PMT) tiene una carga máxima calibrada de algunos cientos de PEs normalmente y que no excede 10^4 PEs puesto que arriba de dicho valor ya no se usan los PMTs debido a que los *afterpulses* pueden intervenir en la medición de los ToT dando medidas falsas. Después de haber aplicado los cortes mencionados a todos los tanques del arreglo, aquellos con al menos un PMT tomando datos son considerados para la posterior reconstrucción.

El error angular y la capacidad de distinguir eventos iniciados por fotones de eventos iniciados por hadrones dependen de la energía y el tamaño de las cascadas al llegar a nivel de tierra. Se adoptan entonces cortes de análisis y una descripción de la resolución angular que dependen del tamaño medido de la cascada. Con estos cortes se divide a los datos en 9 bines, B^3 . El tamaño del evento se define como la razón de los PMT usados para la reconstrucción de los eventos y los PMT disponibles para su reconstrucción, f_{hit} . La figura 3.3 muestra la distribución de energías como función del número de bin para una fuente con espectro de la forma $E^{-2.63}$ mientras que la tabla 3.1 muestra la energía promedio de cada bin. Cabe destacar que esta distribución depende del espectro de la fuente y del ángulo en el que ésta culmina durante su tránsito.

3.2.4. Reconstrucción del núcleo

En una cascada atmosférica existe una concentración mayor de partículas secundarias a lo largo de la trayectoria de la partícula primaria, a esto se le conoce como el núcleo de la cascada y es muy importante su determinación ya que con ello se puede reconstruir la dirección de la partícula primaria que inició la cascada. La figura 3.4 muestra que el núcleo de la cascada se encuentra bien definido en los tanques donde se tiene una mayor cantidad de carga almacenada⁴.

¹Esto debido a que los muones verticales son una fuente casi constante de luz que los vuelve convenientes para establecer la eficiencia total de los PMTs

²Los pulsos en los PMT son amplificados y pasados a través de dos discriminadores a 1/4 y 4 PEs antes de ser digitalizados. El tiempo que pasan los pulsos por encima de los discriminadores (ToT o Time over Threshold) se usa para estimar la carga que se almacenó en cada PMT.

³La manera actual de medir energías de HAWC utiliza una estimación basada en 9 bines. El método usado en esta tesis es diferente y se describe más adelante. Aún no hay una publicación en la que se utilicen estos estimadores de energía

⁴Los colores indican la cantidad de carga almacenada en cada tanque.

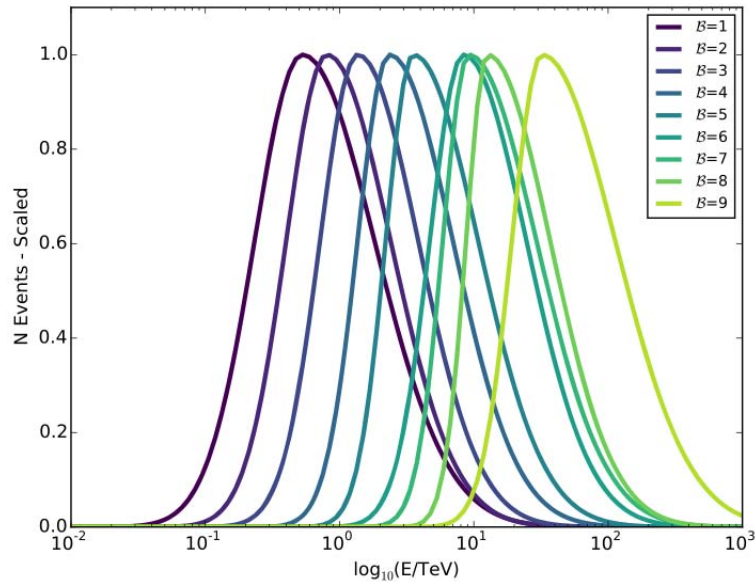


Figura 3.3: Ajustes a la distribución de energía para una fuente con espectro de la forma $E^{-2.63}$ con declinación de $+20^\circ\text{N}$ considerando los bins 1-9. Las distribuciones fueron re-escaladas para tener la misma altura vertical. Tomada de [57].

\mathcal{B}	\tilde{E}_γ^{MC}
1	0.7
2	1.1
3	1.8
4	3.5
5	5.6
6	12
7	15
8	21
9	51

Tabla 3.1: Energía promedio de los 9 bins del análisis estándar de HAWC. \tilde{E}_γ^{MC} representa la energía promedio de los fotones de rayos gamma simulados en cada bin de análisis para una fuente situada a una declinación de 20° con un espectro $E^{-2.63}$ (Una fuente como la Nebulosa del Cangrejo).

La distribución de PEs a nivel de tierra se ajusta con una función que decrece de manera continua con la distancia al núcleo de la cascada. Así, la señal en el i -ésimo PMT será de la forma:

$$S_i = S(A, x, x_i) = A \left(\frac{1}{2\pi\sigma^2} e^{-|x_i - x|^2/2\sigma^2} + \frac{N}{(0.5 + |x_i - x|/R_m)^3} \right), \quad (3.1)$$

donde x es la localización del núcleo, x_i la localización de la medición, R_m es el radio de Molière de la atmósfera, 124.21 m a la altura de HAWC, σ es el ancho de la Gaussiana y N es la normalización de la cola. Los valores usados para σ y N son 10 m y 5×10^{-5} respectivamente. Los parámetros que quedan libres son la ubicación del núcleo y la amplitud A .

Esta función se denominó el *Ajuste Súper Rápido del Núcleo* (SFCF por sus siglas en inglés Super Fast Core Fit) la cual es una simplificación de la función modificada de Nishimura-Kamata-Greisen (NKG) [58] y se escoge por su rapidez para hacer el ajuste de las cascadas atmosféricas. La función NKG toma en cuenta también la edad de la cascada además de evaluar funciones gamma lo cual la vuelve muy costosa computacionalmente. La SFCF hace un tratamiento similar pero realiza una minimización numérica que converge más rápido debido a la simplicidad de la función 3.1, además de que las derivadas se toman analíticamente y no hay polo en la ubicación del núcleo de la cascada. El ajuste se inicializa con un algoritmo que calcula el centro de masa a partir de la carga almacenada en cada PMT.

La figura 3.5 muestra la carga almacenada en cada PMT como función de la distancia, a nivel de tierra,

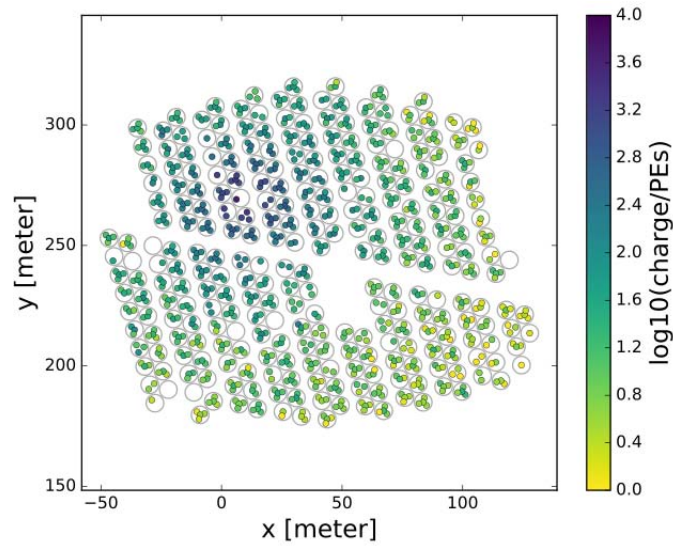


Figura 3.4: Evento de rayos gamma proveniente de la nebulosa del Cangrejo. Se muestra el arreglo HAWC con los círculos grandes representando cada WCD y los círculos más pequeños representando los PMT, en colores la cantidad de luz (medida en PE) registrada en cada PMT. El núcleo de esta cascada puede verse en la zona donde se tiene una mayor densidad de PE. Imagen tomada de [57].

al núcleo reconstruido de la cascada. A diferencia de la función NKG que describe mejor la distribución lateral, la función SFCF permite identificar rápidamente el núcleo de las cascadas. Por ejemplo el núcleo es ubicado con un error promedio de ~ 2 m para eventos grandes ($\mathcal{B} = 8$) y de ~ 4 m para eventos pequeños ($\mathcal{B} = 3$) para cascadas electromagnéticas cuyo núcleo queda dentro del detector HAWC [57]. El error en la ubicación del núcleo incrementa conforme éste queda fuera y más lejos del arreglo del detector. Por ejemplo un núcleo que esté a 50 m del arreglo tendrá error promedio de ~ 35 m en su ubicación.

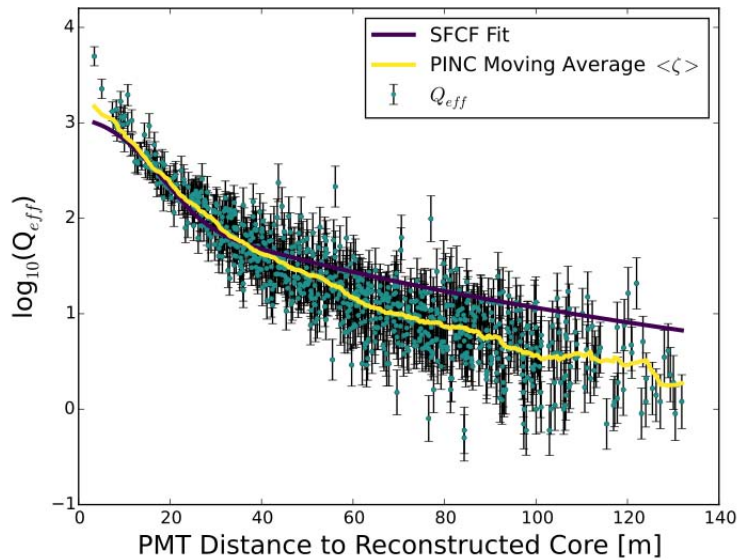


Figura 3.5: Función de distribución lateral, se muestra la carga efectiva registrada por los PMT como función de la distancia al núcleo reconstruido de la cascada. Imagen tomada de [57].

3.2.5. Reconstrucción de la dirección

A primera aproximación se puede suponer que las partículas de la cascada llegan al detector en un plano definido por la velocidad de la luz y la dirección de la partícula primaria. Con un análisis más a fondo es posible ver que de hecho las partículas de la cascada no llegan en un plano, sino más bien el frente de la cascada tiene una curvatura. Las partículas que viajan a lo largo del eje de la cascada viajan menores

distancias que aquellas partículas que viajan lejos de este. Esto se ve reflejado en el detector de tal forma que las partículas cercanas al núcleo de la cascada son detectadas antes que las que están más lejos. La dirección perpendicular al plano de la cascada define la dirección de llegada de la partícula primaria que inició la cascada.

3.3. Separación Gamma/Hadrón

De todas las cascadas atmosféricas detectadas en HAWC, las más abundantes (100000 hadrones por cada fotón) son las producidas por rayos cósmicos y constituyen la fuente principal de fondo en la observación de fotones de altas energías. Las cascadas atmosféricas producidas por rayos cósmicos y rayos gamma tienen huellas distintivas que permiten diferenciarlas entre sí. Por ejemplo, una cascada electromagnética tiene un núcleo bien definido y una distribución de carga suave en función de la distancia; algo que no ocurre con las cascadas hadrónicas, las cuales son ricas en piones, muones, etc. y que al producirse suelen tener alto momento transversal. La figura 3.6 es un ejemplo de la huella que deja una cascada hadrónica y una electromagnética al llegar al arreglo de HAWC.

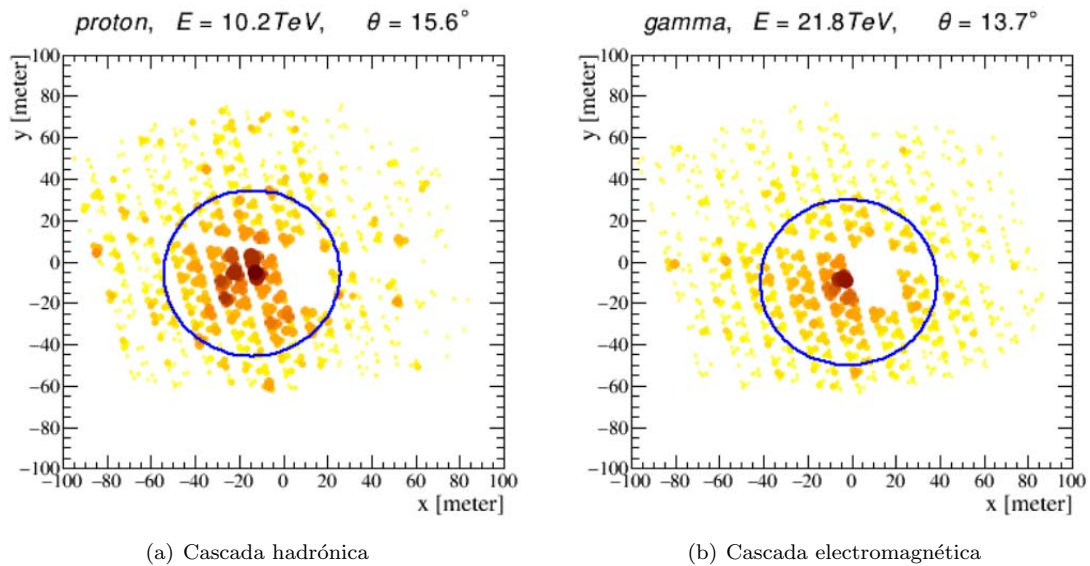


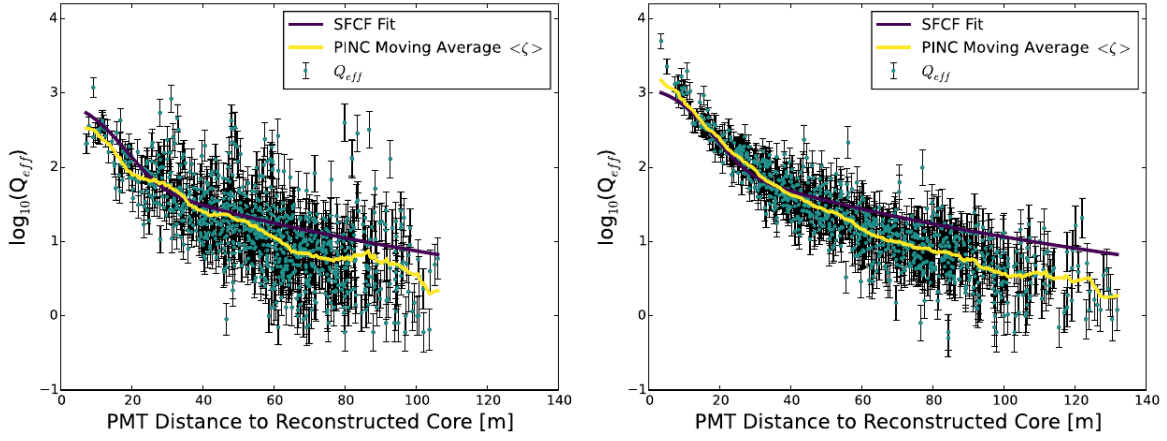
Figura 3.6: Comparación de una cascada hadrónica (izq.) y una electromagnética (der.) vistas con HAWC. Ambas cascadas tienen un núcleo definido pero la cascada hadrónica presenta una mayor dispersión con relación a la electromagnética. Tomada de [54].

La figura 3.7 muestra la distribución lateral de carga para dos tipos de cascadas, una hadrónica y otra electromagnética con gran probabilidad de haber sido producida por un fotón proveniente de la nebulosa del Cangrejo. Es posible ver cómo en ambas cascadas la carga efectiva en promedio disminuye para aquellos hits que quedan más alejados del núcleo reconstruido. Nótese también que en la cascada hadrónica existen hits con cargas efectivas grandes en las partes más lejanas del núcleo reconstruido. Esta *grumosidad* es característica de cascadas hadrónicas y resulta de una combinación de partículas penetrantes (muones principalmente) y algunas otras sub-cascadas que no suelen estar presentes en las cascadas electromagnéticas.

Para la identificación y separación de las cascadas hadrónicas que constituyen el fondo de rayos cósmicos se utilizan dos parámetros. La *compacidad* (en inglés *compactness*) busca identificar PMTs con cargas efectivas grandes que se encuentren lejos del núcleo de la cascada y se define como sigue:

$$compactness = \frac{nHitSP20}{CxPE40}, \quad (3.2)$$

Donde $CxPE40$ es el valor de la mayor carga efectiva que se registró en un PMT fuera de un radio de 40 m del núcleo de la cascada y se mide en fotoelectrones y $nHitSP20$ es el número total de PMTs disparados dentro de una ventana de 20 ns a partir del contacto con el frente de la cascada.



(a) Distribución lateral de carga para una cascada hadrónica (izq.) (b) Distribución lateral de carga para una cascada electromagnética (der.)

Figura 3.7: Distribución lateral de carga para una cascada hadrónica (izq.) y una electromagnética producida por un fotón proveniente de la nebulosa del Cangrejo (der.). La cascada hadrónica producida por un rayo cósmico tiene grandes cargas aisladas lejos del núcleo; característica que no está presente en la cascada electromagnética. Imagen adaptada de [57]

Como se observa en la figura 3.6, elegir 40 m asegura que la región nuclear y su gran concentración de hits quede excluida de la variable CxPE40. De esta forma las cascadas hadrónicas que tienen un mayor valor de CxPE40 debido a su gran dispersión en todo el arreglo, estarán caracterizadas por un menor valor de la *compacidad* que las cascadas de rayos gamma. Cabe destacar que para poder realizar esta separación es necesario que el ajuste en la posición del núcleo de la cascada haya sido exitoso [54].

El otro parámetro utilizado para la separación mide la *grumosidad* de la cascada; se denomina PINC (del inglés Parameter for the Identification of Nuclear Cosmic rays) y se identifica con la letra \mathcal{P} . El valor de \mathcal{P} se calcula utilizando la función de distribución lateral, vista en la figura 3.7. Como vimos, cada PMT registra una carga efectiva Q_{eff} , \mathcal{P} se calcula utilizando el logaritmo de dicha carga efectiva $\zeta_i = \log_{10}(Q_{eff,i})$. A cada hit se le asigna un valor esperado $\langle \zeta_i \rangle$ promediando ζ_i en todos los PMT dentro de un anillo centrado en el núcleo de la cascada, de ancho 5 m que contiene al hit. \mathcal{P} entonces se calcula usando la ecuación de χ^2 :

$$P = \frac{1}{N} \sum_{i=0}^N \frac{(\zeta_i - \langle \zeta_i \rangle)^2}{\sigma_{\zeta_i}^2} \quad (3.3)$$

Los errores σ_{ζ_i} se asignan mediante el estudio de una muestra de fuertes candidatos de rayos gamma en la vecindad de la Nebulosa del Cangrejo.

La variable \mathcal{P} describe la suavidad axial. La línea amarilla de la figura 3.7 muestra como se distribuye el promedio ζ_i de ambos eventos. De acuerdo con dicha figura, el evento hadrónico es *grumoso* y tiene hits que difieren mucho de la tendencia promedio por lo que su valor de \mathcal{P} sería grande.

3.4. Estimación de la Energía

Dada la dificultad y baja resolución en la estimación de la energía utilizando un arreglo en la superficie, la colaboración HAWC ha desarrollado dos algoritmos independientes para la estimación de la energía de los rayos gamma. Los métodos son conocidos como Ground Parameter (GP) y Neural Network (NN) los cuales se describirán a continuación. Como notación \hat{E} se refiere a la energía estimada mientras que E se refiere a la energía real tomada de la simulación.

3.4.1. Ground Parameter

Este método se basa principalmente en la densidad de carga hasta una distancia óptima del núcleo de la cascada. Es una modificación a pequeñas escalas del método utilizado para experimentos grandes de rayos cósmicos como Pierre Auger.

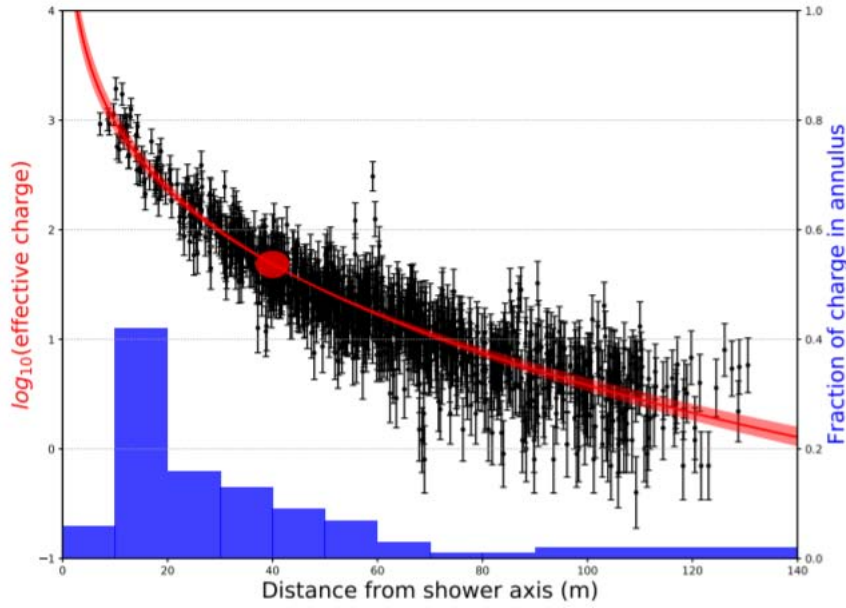


Figura 3.8: Los puntos negros muestran el logaritmo de la carga efectiva como función de la distancia al eje de la cascada. La línea roja es el mejor ajuste a una función tipo NKG, mientras que la banda roja se usó para determinar el radio óptimo. El histograma azul es la fracción de carga en diversos anillos, esta información se usa como entrada para el estimador NN. Imagen tomada de [59]

El radio en el que se minimiza la incertidumbre de la densidad de energía es a lo que se le define como *radio óptimo*. Este radio debe estar lo suficientemente lejos del núcleo de la cascada para evitar las fluctuaciones entre cascadas que hacen que la estimación sea complicada, pero también debe estar lo suficientemente cerca del eje de la cascada de tal forma que la señal de los PMT sea lo suficientemente grande para que las fluctuaciones no sean tan importantes. La determinación de este radio óptimo se utiliza una versión modificada de la función NKG [58], incluyendo un factor de $1/r$ para medir densidad de energía [55] en lugar de densidad de partículas.

$$\log_{10}(NKG/r) = A + s \left[\log_{10} \frac{r}{r_m} + \log_{10} \left(1 + \frac{r}{r_m} \right) \right] - 3 \log_{10} \frac{r}{r_m} - 4.5 \log_{10} \left(1 + \frac{r}{r_m} \right) \quad (3.4)$$

Aquí, A es el logaritmo de la amplitud del ajuste, s está relacionado con la edad de la cascada y r_m es el radio de Molière el cual es 124.21 m para HAWC, r es la distancia del PMT al eje de la cascada. A y s son los únicos dos parámetros libres del ajuste. La figura 3.8 muestra un ejemplo del ajuste de una función de distribución lateral a la función NKG/r . Una vez que se ha obtenido el mejor ajuste, s se varía un $\pm 10\%$ y se aplican ajustes adicionales dejando la normalización libre. Este procedimiento genera una banda de ajustes (figura 3.8). El punto donde el ancho de la banda es más pequeño es el punto donde se minimiza la incertidumbre de no conocer la forma de la función de distribución lateral. Este es el valor del radio óptimo. Para HAWC este valor es 40 m y no depende del zenith ni de la energía de la partícula primaria.

La ecuación 3.4 se evalúa entonces para $r = 40$ m y se transforma a energía. A este valor se le conoce como $\log_{10}s_{40}$. La implementación más simple sería usar la simulación Monte Carlo para realizar un ajuste de la forma $\log_{10}\hat{E} = f(s_{40})$. Ahora, la señal medida a nivel de piso varía de manera significativa con la inclinación, es decir con el ángulo zenital, debido a la diferente profundidad de atmósfera que deben atravesar las cascadas atmosféricas. Entonces la fórmula anterior se parametriza para tener en cuenta el ángulo zenital: $\log_{10}\hat{E} = f(s_{40}, \theta)$. La forma funcional exacta se escoge de manera empírica para empatar con los eventos simulados.

$$\log_{10}\hat{E} = m(\theta)\log_{10}s_{40} + c(\theta) \quad (3.5)$$

3.4.2. Neural Network

El algoritmo de reconstrucción de energía con NN utiliza una red neuronal para estimar las energías de los fotones primarios a partir de variables calculadas como parte de la reconstrucción en HAWC. El estimador de energía NN utiliza una arquitectura multicapa (*multilayer-perceptron*) con dos capas ocultas y una función

de activación logística. Las capas contienen 15 y 14 nodos en la primera y segunda respectivamente. Los valores de los 479 *pesos* se escogen para minimizar la función de error.

$$D(w) \equiv \frac{1}{2} \sum_{i=1}^2 u_i \left[\log_{10} \hat{E}(x_i; w) - \log_{10} E_i \right]^2 \quad (3.6)$$

evaluada usando eventos Monte Carlo, donde w es el vector de pesos de la red neuronal, u_i es la importancia relativa del evento i , \hat{E} es la función que devuelve un estimado de la energía para un vector dado de energía y de peso y E_i es la energía real del evento i extraída de la simulación Monte Carlo. Las u_i se escogen de tal forma que se recupere un espectro de ley de potencias E^{-2} , el cual se selecciona ya que una red neuronal entrenada sobre este espectro produce un error RMS relativo constante para energías entre 1 y 100 TeV.

Las variables de entrada de la red neuronal se escogen de tal manera que caractericen tres grandes propiedades de una cascada atmosférica: la cantidad de energía depositada en el detector, la extensión de la cascada que queda contenida en el detector y el grado de atenuación de la cascada debido a la atmósfera.

Para inferir la cantidad de energía depositada en el detector se usan tres cantidades: la fracción de PMTs que tuvieron activados en un evento, la fracción de tanques activados y el logaritmo base 10 de la normalización obtenida del ajuste de la función de distribución lateral. Estos parámetros están correlacionados con la energía de la partícula primaria que inició la cascada.

La fracción de la cascada que cae dentro del detector se infiere usando la distancia entre el núcleo reconstruido y el centro del arreglo de HAWC.

3.5. Variables y tipos de archivos

Ya que se ha hecho la reconstrucción de las cascadas atmosféricas (o eventos), toda la información recabada es almacenada en archivos de tipo XCDF⁵ (The The eXplicitly Compacted Data Format por sus siglas en inglés) para su posterior análisis. Actualmente, de cada evento se guardan alrededor de 64 columnas correspondientes a diferentes variables (tabla 3.2). Los datos son recabados por día en directorios llamados *runs* o corridas. Cada corrida se divide en *subruns* (sub-corridas) que son archivos XCDF que guardan todos los eventos dentro de una ventana de 2 minutos (recordemos que la tasa de eventos es de 24 KHz). Entonces cada *run* contiene 692 *subruns*.

rec.status	rec.version	rec.eventID	rec.gpsSec
rec.runID	rec.timeSliceID	rec.event_flags	rec.nChAvail
rec.nHit	rec.trigger_flags	rec.gtc_flags	rec.nHitTot
rec.dec	rec.nHitSP10	rec.gpsNanosec	rec.nTankHit
rec.ra	rec.nHitSP20	rec.nTankTot	rec.planeChi2
rec.coreX	rec.nTankHitTot	rec.nTankAvail	rec.SFCFChi2
rec.coreY	rec.windowHits	rec.SFCFNDOF	rec.gammalheLLH
rec.logMaxPE	rec.angleFitStatus	rec.CxPE40XnCh	rec.protonlheLLH
rec.logNPE	rec.coreFitStatus	rec.coreFiduScale	rec.chargeFiduScale50
rec.CxPE40	rec.logCoreAmplitude	rec.fAnnulusCharge0-8	rec.chargeFiduScale70
rec.PINC	rec.protonlheEnergy	rec.gammalheEnergy	rec.chargeFiduScale90
rec.CxPE40SPTIME	rec.nChTot	rec.coreFitUnc	rec.planeNDOF
rec.zenithAngle	rec.azimuthAngle	rec.CxPE40PMT	rec.LDFAge
rec.logNNEnergy	rec.LDFamp	rec.LDFChi2	rec.disMax

Tabla 3.2: Variables contenidas en los *subruns*. El número de eventos en cada *subrun* es de aproximadamente 4 millones.

Como ya se mencionaba, las partículas que se simulan también pasan a través del algoritmo de reconstrucción. Entonces la información que contienen los archivos de simulación es en principio la misma que los archivos de datos. Lo que los hace diferentes es que además contienen información relevante con la que se inicia la simulación, por ejemplo la partícula primaria que inicia la cascada, su energía inicial y su dirección entre otras cosas (tabla 3.3). Este tipo de archivos se obtiene simulando de manera individual distintos tipos de partículas primarias, generando así distintos archivos de simulación que después se combinan en uno sólo conocido como *sweets* (del inglés Software for Weighting Events and Eventlike Things and Stuff).

⁵Además de comprimir de manera sustancial los archivos, esta paquetería permite almacenar datos con una precisión definida por el usuario.

mc.corsikaParticleId	mc.status	mc.prescale	sweets.oneWgt
mc.coreFiduScale	sweets.IWgt	sweets.TWgt	sweets.BWgt
mc.logGroundEnergy	mc.radiusWeight	mc.eventWeight	mc.zenithAngle
mc.azimuthAngle	mc.delCore	mc.delAngle	mc.coreR
mc.logEnergy	mc.coreX	mc.coreY	

Tabla 3.3: Además de las variables de la tabla 3.2 los archivos de simulación contienen información referente a los parámetros iniciales de la simulación. El número de eventos es alrededor de 18 millones y contiene fotones, protones entre otros núcleos.

Variable	Descripción
rec.nChAvail	Canales usados después de la calibración
rec.nHitTot	Hits que pasaron 2 y 4 umbrales
rec.nHit	Hits después de la selección
rec.nHitSP10	Número de hits dentro de 10 nanosegundos después del frente de la cascada
rec.nHitSP20	Número de hits dentro de 20 nanosegundos después del frente de la cascada
rec.nTankHitTot	Número de tanques con PMTs en funcionamiento en el subrun
rec.nTankHit	Número de tanques con eventos después de la selección
rec.windowHits	Número total de hits dentro de una ventana de 150 ns
rec.PlaneNDOF	Grados de libertad del ajuste Gaussiano del plano de la cascada
rec.SFCFNDOF	Grados de libertad del ajuste SFCF. Se usa para determinar coreX y coreY
rec.CxPE40PMT	ID del PMT con el máximo número de fotoelectrones >a 40 m
rec.CxPE40XnCh	Número de canales excluidos <40 m
rec.coreFiduscale	Parámetro para distinguir si una cascada aterrizó dentro o fuera del detector
rec.zenithAngle	Ajuste del ángulo zenital
rec.azimuthAngle	Ajuste del ángulo azimutal
rec.planeChi2	χ^2 del ajuste del plano
rec.coreX	Ajuste en X del núcleo
rec.coreY	Ajuste en Y del núcleo

Tabla 3.4: Descripción breve de las variables utilizadas para el entrenamiento de la red neuronal.

Las variables utilizadas para el análisis se describen brevemente a continuación (Tablas 3.4 y 3.5).

En la siguiente sección se da un resumen de las búsquedas de GRBs utilizando 18 meses de datos del observatorio HAWC

3.6. Búsquedas de GRBs

El observatorio HAWC reportó en 2017 la búsqueda de 64 GRBs que fueron detectados por *Swift* y *Fermi*, dentro de esos 64 se incluyen 3 que fueron detectados también por *Fermi-LAT*. A continuación se incluyen los resultados de dicha búsqueda en la que no se logró detectar ningún GRB. Más información sobre el método de búsqueda y la toma de datos pueden encontrarse en [60].

- Durante los primeros 18 meses de funcionamiento del detector, se pudieron tomar datos para estudiar 72 de 76 GRBs en el campo de visión.
- Debido a algunos problemas con el detector en 2015, no se pudieron estudiar 9 destellos. Los destellos que ocurrieron en 2016 si pudieron ser estudiados.
- 4 GRBs detectados por *Fermi-LAT* estuvieron en el campo de visión de HAWC (160503A, 160310A, 150416A y 150314A).

Variable	Descripción
rec.logCoreAmplitude	Log de la amplitud del ajuste del núcleo
rec.coreFitUnc	Error en la posición del ajuste
rec.SFCFChi2	chi2 para el ajuste del núcleo con SFCF
rec.fAnnulusCharge0-8	de 0 a 8, 9 variables
rec.logMaxPE	Máximo de PEs calculado
rec.logNPE	Total de PE calculado
rec.CxPE40	Carga máxima en PE fuera de un radio de 40 m
rec.CxPE40SPTime	delta t del PMT en CxPE40 con respecto al frente de la cascada
rec.LDFAge	Edad de la cascada a partir del ajuste de la distribución lateral
rec.LDFamp	Amplitud a partir del ajuste de la distribución lateral
rec.LDFChi2	chi2 del ajuste a la distribución lateral
rec.PINC	PINCness, separador G/H
rec.disMax	...

Tabla 3.5: Descripción breve de las variables utilizadas para el entrenamiento de la red neuronal (continuación).

- El GRB 160503A no pudo ser observado, debido a problemas con el detector.
- El GRB 150416A sólo fue localizado por *Fermi*-GBM así que fue tratado como un destello de GBM en el análisis.
- El GRB 150314A que también disparó a *Swift* ocurrió cuando la toma de datos de HAWC era inestable. Se hizo una inspección mediante mapas de significancia de los datos no hallando nada significativo.
- Además del GRB 150314A se encontraron 21 destellos de *Swift* en el campo de visión de HAWC.
 - Cuando ocurrió el GRB 150910A la tasa de adquisición de datos del detector era inestable por lo que no se pudo hacer análisis de los datos.
 - El GRB 150323A ocurrió cuando la toma de datos de HAWC era inestable así que no hubo datos disponibles para un análisis. Sin embargo se hizo una inspección de los mapas de significancia que no reveló señal alguna.

Los dos GRBs anteriores son los que tienen el más largo T_{90} y, como HAWC no es tan sensitivo a GRBs largos, es de esperarse que los GRBs no se hayan detectado.

- La fase temprana del GRB 160624A no se observó con HAWC debido a problemas con el detector.
- Durante el mismo periodo *Fermi*-GBM detectó 64 GRBs que estuvieron en el campo de visión de HAWC, lo cual incluye las cuatro detecciones de *Fermi*-LAT mencionadas anteriormente y las 9 co-detecciones con *Swift* (además del GRB 150314A), lo cual deja 51 GRBs observados solamente por *Fermi*-GBM.
 - No se tienen datos en HAWC para bn 160609690 and bn 150520893.
 - Para bn 151211672, bn 150911315, bn 150819440, bn 150612702, bn 150528656 y bn 150208929 el detector mostraba inestabilidades por lo que tales destellos no pudieron ser analizados.
- El GRB 170206A es el tercer GRB corto más brillante detectado por *Fermi*-GBM
- Todos los GRBs son consistentes con la suposición de sólo fondo.
- Sólo fue posible reportar límites superiores.

En esta primera etapa los algoritmos de HAWC aún no contaban con optimizadores para detectar fotones de cientos de GeV, algo en lo que contribuye este trabajo de tesis además de otras mejoras hechas por otros grupos de colaboradores. Con estas mejoras se espera pronto detectar un GRB con HAWC.

Capítulo 4

Creación de Mapas

El proceso de creación de mapas de significancia es un proceso que requiere varias etapas. En este capítulo se describe de manera breve dicho proceso y algunos ejemplos de los resultados.

4.1. LiFF

En HAWC se desarrolló un método para análisis de datos que trabaja a nivel de eventos. Este software fue desarrollado de tal forma que lo único que necesita el usuario es introducir el modelo físico que se desea poner a prueba. LiFF (por sus siglas en inglés, Likelihood Fitting Framework)[61] permite el análisis estadístico utilizando un formalismo de verosimilitud (likelihood). Las distintas interfaces de LiFF son: un modelo físico, detalles sobre la respuesta del detector, los datos separados por bins y el cálculo de la verosimilitud.

4.1.1. Formalismo de verosimilitud

El propósito principal de LiFF es facilitar al usuario el cálculo de la verosimilitud de un modelo físico en particular y estimar una respuesta del detector dado un conjunto de datos observados. La forma en que el esquema de trabajo se reduce es mediante el *bineado* de los datos, es decir, se calcula la verosimilitud para cada bin de datos. Los datos de las cascadas en HAWC se encuentran separados por bins en un arreglo multi-dimensional de atributos de cada evento. En un análisis típico de HAWC se incluye la dirección de los eventos en un conjunto de coordenadas (por ejemplo ascensión recta y declinación), la calidad de los eventos y/o una medida de su energía.

Computacionalmente es menos costoso calcular el logaritmo de la verosimilitud:

$$\ln\mathcal{L}(\theta; N_{obs}) = \sum_{i=0}^{Bins} \ln(f((N_{obs})_i|\theta)), \quad (4.1)$$

donde θ es un conjunto de parámetros, $(N_{obs})_i$ es el número de eventos en el i -ésimo bin, N_{obs} es la colección de todos los $(N_{obs})_i$ y f es la probabilidad de observar $(N_{obs})_i$ dada una θ . Para estimar el conjunto óptimo de parámetros θ_0 , se encuentra el θ que maximiza el $\ln\mathcal{L}$ o, de manera equivalente, se minimiza el valor de $-\ln\mathcal{L}$.

Para comparar hipótesis, se utiliza el cociente de verosimilitud (*Likelihood Ratio*). Un test estadístico se define como:

$$TS = 2\ln \frac{\mathcal{L}(HA; N_{obs})}{\mathcal{L}(HN; N_{obs})}, \quad (4.2)$$

donde HA es una hipótesis alternativa y HN una hipótesis nula. En una hipótesis nula uno supone que lo observado es producto de fluctuaciones del fondo mientras que en la hipótesis alternativa se supone la existencia de una fuente con una normalización del flujo, X. Cuando la hipótesis nula es verdadera y en el caso de modelos anidados¹, TS está distribuida de manera aproximada como una χ^2 con el número de grados de libertad igual a la diferencia en el número de parámetros libres entre las hipótesis. En este caso la diferencia entre ambas hipótesis da un sólo parámetro, la normalización del flujo X. Si la hipótesis nula es cierta (en el caso de que no hay fuentes) TS se distribuye como $\chi^2(NDOF = 1)$, y \sqrt{TS} puede interpretarse como la significancia de la *sobre-abundancia* de eventos, en unidades de *Sigmas-Gaussianas* [61].

¹Un modelo A se encuentra anidado en un modelo B si los parámetros de A son un subconjunto de los parámetros de B

El conjunto de datos

El software LiFF es un algoritmo de análisis a nivel de eventos, es decir, cascadas atmosféricas reconstruidas para HAWC. Los datos correspondientes a eventos son separados en bins en el cielo usando pixeles de HEALPix [62] que, básicamente son pixeles que ocupan áreas iguales. Se le conoce como mapa de datos al conjunto de pixeles de HEALPix que cubren todo el cielo. A nivel de eventos, los datos pueden separarse en bins usando otras dimensiones. A manera de ejemplo, en la colaboración HAWC los datos se separan en 10 bins que miden la calidad de los eventos². Para este trabajo sólo se utilizó un solo bin, el cual se describirá en próximas secciones.

El modelo físico

El modelo físico se implementa como un conjunto de modelos de una o más fuentes. En este sentido se definen dos tipos de fuentes: 1) Fuente puntual y 2) Fuente extendida. Las fuentes puntuales tienen una ubicación además de un espectro mientras que las fuentes extendidas tienen ubicación en el cielo, espectro y una morfología espacial. Nótese que cada fuente puede tener una enorme cantidad de parámetros libres al momento de definir el modelo físico.

Modelos de respuesta del detector

Dada la declinación de una fuente en el cielo y un espectro arbitrario para la fuente, el código encargado de la respuesta del detector calcula las PSF (Point Spread Function)³ esperadas así como el número de eventos en cada bin de análisis. La información obtenida se usa para alimentar el modulo de que calcula la verosimilitud del modelo para así obtener el número de eventos no hadrónicos en cada bin para un modelo físico en particular. La manera en que se describe la respuesta del detector es mediante histogramas en una dimensión. Dichos histogramas se generan a partir de la simulación usando un espectro en particular. El software LiFF tiene la ventaja de que si el espectro llamado para el modelo físico difiere del espectro usado para la respuesta del detector, entonces ésta respuesta se modifica re-pesando los histogramas de distribución de energía en cada bin. Este método funciona bien siempre y cuando el espectro del modelo físico y el de la respuesta del detector no difieran demasiado.

4.1.2. Cálculo de la verosimilitud

Para calcular $\ln\mathcal{L}$ primero se necesita saber el número esperado de eventos correspondientes a señales y el número esperado de eventos de fondo para cada bin. El número esperado de eventos de señal en cada bin relevante se calcula a partir del modelo introducido y la información contenida en la respuesta del detector. El número esperado de eventos de fondo (eventos de hadrones que pasaron los cortes de calidad) no se calculan con la simulación sino más bien con los mismos datos (esto porque en los datos se tienen enormes cantidades de eventos de hadrones). El fondo puede estimarse en el momento en el que los datos se separan en bins o bien puede estimarse como un parámetro extra como parte de la minimización de $-\ln\mathcal{L}$

Usando una distribución de probabilidad de tipo Poisson el $\ln\mathcal{L}$ se calcula como:

$$\ln\mathcal{L} = \sum^{Bines} (N_{obs}\ln N_{exp} - N_{exp} - \ln(N_{obs}!)), \quad (4.3)$$

donde, como se mencionaba, N_{obs} es el número de eventos observados y N_{exp} es el número de eventos esperados.

Map Maker

El software de creación de mapas de HAWC se conoce como *map maker*. Su propósito es crear mapas de todo el cielo a partir de las cascadas atmosféricas reconstruidas. Los mapas constan de histogramas bidimensionales de las direcciones de llegada de las cascadas. Como se sabe, sólo una pequeña cantidad de las cascadas atmosféricas corresponde a fotones por lo que el fondo puede estimarse directamente de los datos recabados. La manera en que se calcula el fondo es mediante *integración directa* [63] que es una técnica que convolucionan la distribución local de los eventos con su distribución de tiempos en intervalos de 2 horas en el caso de búsquedas de fuentes puntuales.

²Los eventos en un mismo bin tienen resolución angular similar y una separación G/H también similar, lo que hace que la respuesta del detector sea función de las características de los eventos. La calidad de eventos, como ya se observó, es el número de PMTs activados dentro del arreglo, lo cual está relacionado con la energía depositada en el arreglo

³Debido a las limitaciones de un detector, los eventos en el cielo tienden a verse extendidos al ser observados en el detector

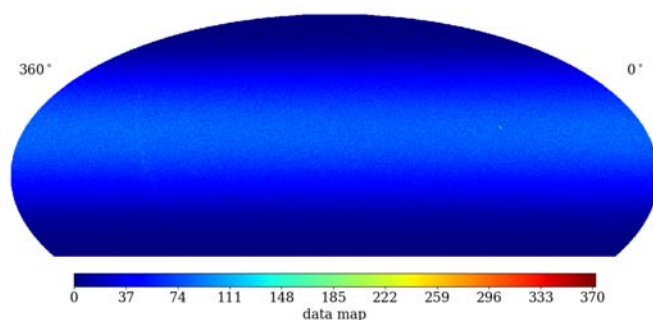
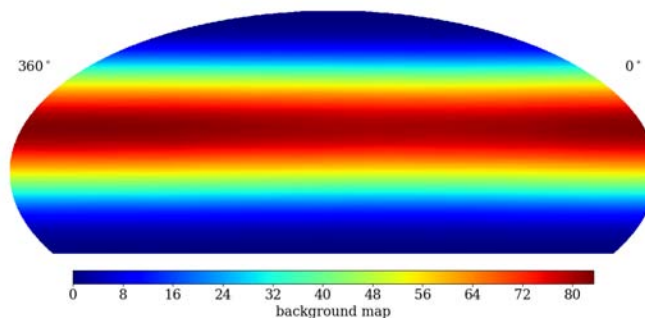
(a) Mapa de datos generado con `make-hawc-maps`(b) Mapa de fondo generado con `make-hawc-maps`

Figura 4.1: Mapas de datos y fondo de todo el cielo. La barra de color indica el número de eventos.

El módulo `map maker` utiliza archivos reconstruidos cuyos eventos (cascadas atmosféricas) están ordenados de manera cronológica. Con estos construye un archivo de mapas que contiene los seis mapas necesarios para un cálculo de significancia en todo el cielo. La sintaxis para la creación de mapas es la siguiente:

```
aerie -apps -make-hawc-maps --input reco_run007084_*.xcd --cutfile cortes.txt \
--nSide 512 --useJ2000 --roi \
--dtMax_hr 0.9 --dtMin_hr 0.1 -n map_512
```

El programa devuelve una serie de archivos FITS (Flexible Image Transport System), cada uno contiene 6 mapas de HEALPix (cada uno en una de las columnas del archivo FITS) necesarios para el cálculo de la significancia, contiene un mapa de datos y un mapa del fondo estimado a partir de los eventos. El número de archivos FITS generados depende de los cortes de selección que hay dentro del archivo `cortes.txt`. La figura 4.1 muestra cómo se ve un mapa generado con `make-hawc-maps`:

4.2. Respuesta del detector

El comando para generar una respuesta del detector es el siguiente:

```
liff -MakeDetectorResponse -s ../SWEETS/ -c cortes.txt \
-d -35,-30,-25,-20,-15,-10,-5,0,5,10,15,20,25,30,35,40,45 \
-o liff_det_res.root
```

La respuesta del detector es una serie de histogramas de la PSF y de la distribución de energía. Existen dos histogramas por cada bin que se analiza y por cada intervalo de declinación.

4.3. Mapas de significancia

El comando para la generación de un mapa de significancia es:

```
aerie -apps -HealpixSigFluxMap \
-i crab_strip-run007060_allsub_NN0.6_25Feb_bin0NN_N512.fits.gz \
-b 0NN -d liff_det_res_NN0.6.root \
-o sigmap_crabstrip-run007060_allsub_NN0.6_25Feb_bin0NN_N512.fits.gz \
-s 'SimplePowerLaw,3.5e-11,2.63' --pivot 7 --padding 1
```


Los mapas de significancia son el resultado final. Los excesos visibles en este tipo de mapas permiten identificar posibles fuentes de rayos gamma ya que el fondo suele estar conformado por rayos cósmicos. Todo aquello que tenga significancia menor a 5σ es considerado una fluctuación del fondo de rayos cósmicos mientras que por arriba de 5σ existe la certeza de que lo observado no es una fluctuación del fondo.

En el siguiente capítulo se da una introducción de lo que es el análisis de componentes principales y su importancia en la realización de este trabajo.

Capítulo 5

Análisis de Componentes principales

Las componentes principales son un conjunto de nuevas variables obtenidas a partir de los datos originales. Éstas se caracterizan por no tener correlación alguna entre sí y además se ordenan de acuerdo con la información que llevan incorporada. Una manera de medir la cantidad de información contenida en una componente es mediante la varianza, es decir, cuanto mayor sea su varianza mayor es la cantidad de información contenida en dicha componente. Para que el Análisis de Componentes Principales (ACP) tenga el efecto esperado, las variables originales deben estar correlacionadas, si no existiera correlación alguna las componentes principales serían igual a las variables originales.

El ACP es una herramienta puramente estadística¹ que permite obtener, a través de los valores y vectores propios de la matriz de correlaciones (de la cual se hablará en las siguientes secciones), las componentes que describan la máxima varianza del conjunto de datos. Las primeras secciones de este capítulo tratan algunos conceptos matemáticos y estadísticos, además de algunos ejemplos, con los que se pueda entender el procedimiento y la obtención de las componentes principales a través del ACP, el cual se introduce en la sección final del capítulo. Cabe destacar que, dado que no es el propósito de esta tesis, los conceptos introducidos se dejarán sin demostrar.

5.1. Desviación estándar, varianza y covarianza

Considérese un conjunto de datos cualquiera, para el cual sabemos que siempre es posible hacer estudios estadísticos, como obtener su promedio. Los conjuntos [0, 8, 12, 20] y [8, 9, 11, 12] tienen el mismo promedio (10) y sin embargo son conjuntos completamente diferentes; es entonces evidente que el promedio no brinda información suficiente acerca de las componentes de un conjunto. Si se quiere ver la diferencia en ambos conjuntos se puede analizar que tan dispersos están respecto a su promedio y la manera de hacer esto es mediante la desviación estándar:

$$f = \sqrt{\frac{\sum_{i=1}^N (X_i - \bar{X})^2}{N - 1}} \quad (5.1)$$

Aplicando la ecuación 5.1 a nuestros dos conjuntos iniciales se ve que para el primero $f = 8.3266$ mientras que para el segundo $f = 1.8257$, este resultado es de esperarse ya que el primer conjunto está más disperso con respecto al promedio. Ahora considérese este otro conjunto: [10, 10, 10, 10] su promedio es 10 pero su desviación estándar es cero ya que ningún miembro del conjunto se encuentra alejado del promedio.

Elevando al cuadrado la desviación estándar se obtiene la varianza:

$$f^2 = \frac{\sum_{i=1}^N (X_i - \bar{X})^2}{n - 1} \quad (5.2)$$

Nótese que aunque la varianza no es comúnmente usada, proporciona la misma información que la desviación estándar. En las siguientes secciones se verá la razón de introducir la varianza en lugar de usar la desviación estándar.

Hasta ahora, y para simplificar la comprensión de los conceptos, se han usado conjuntos en una dimensión. Bajo otro nombre, la varianza y la desviación estándar (las cuales operan en una dimensión) tienen aplicación a más dimensiones y esto es lo que nos interesa; saber es si existe alguna relación entre dimensiones. Una

¹Los resultados obtenidos con este análisis carecen de significado físico y pueden tener diversas interpretaciones.

medida similar para ver cómo una dimensión se desvía de su promedio con respecto a las otras dimensiones y sus correspondientes promedios es la covarianza. La covarianza se mide siempre entre dos dimensiones y la covarianza de una dimensión consigo misma es la varianza. La covarianza se define como:

$$\text{cov}(X, Y) = \frac{\sum_{i=1}^N (X_i - \bar{X})(Y_i - \bar{Y})}{N - 1} \quad (5.3)$$

Nótese que cambiar el término $(Y_i - \bar{Y})$ por $(X_i - \bar{X})$ significa comparar la dimensión X consigo misma obteniendo así la varianza de esa dimensión. Además por las propiedades de la multiplicación se tiene que $\text{cov}(X, Y) = \text{cov}(Y, X)$.

5.1.1. Matriz de covarianza y de correlación

Cuando el conjunto de datos contiene un número de dimensiones mayor a dos, pueden entonces hacerse más cálculos de la covarianza. Por ejemplo en 3 dimensiones es posible calcular $\text{cov}(x, y), \text{cov}(y, z), \text{cov}(x, z)^2$. Una forma para tener todas las combinaciones de covarianza entre todas las dimensiones es calcular dimensión a dimensión y colocar todos los valores en una matriz; la matriz de covarianza. Para un conjunto de datos n-dimensional la matriz de covarianza se define como:

$$C^{n \times n} = (c_{i,j}, c_{i,j} = \text{cov}(\text{Dim}_i, \text{Dim}_j)), \quad (5.4)$$

donde $C^{n \times n}$ es una matriz con n filas y n columnas y Dim_j es la j-ésima dimensión. Lo anterior puesto en palabras dice que si se tiene un conjunto de datos con n dimensiones, la matriz de $n \times n$ es el resultado de calcular la covarianza dimensión a dimensión por separado. Retomando el ejemplo en 3 dimensiones, la matriz de covarianza quedaría de la siguiente forma:

$$\begin{pmatrix} \text{cov}(x, x) & \text{cov}(x, y) & \text{cov}(x, z) \\ \text{cov}(y, x) & \text{cov}(y, y) & \text{cov}(y, z) \\ \text{cov}(z, x) & \text{cov}(z, y) & \text{cov}(z, z) \end{pmatrix}$$

Y como ya se había visto, al ser la covarianza simétrica también la matriz de covarianza será simétrica y los valores de la diagonal estarán dados por la covarianza de cada dimensión consigo misma; su varianza.

Generalmente los datos reales vienen expresados en diferentes unidades: masas, distancias, luminosidades etc. que pueden tener diferentes ordenes de magnitud. Para trabajar con este tipo de conjuntos de datos es habitual primero estandarizarlos, es decir, hacer que el promedio y la varianza de cada dimensión sean igual a cero y uno respectivamente. La matriz de covarianza con datos estandarizados lleva por nombre matriz de correlaciones. En la matriz de correlaciones, la diagonal está compuesta por unos y sus valores y vectores propios son distintos a la matriz de covarianza. Es por ello que se hace la distinción entre una matriz y la otra.

5.2. ACP, el método

Como ya se mencionaba, el ACP requiere del cálculo de valores y vectores propios de la matriz de correlaciones. Utilizando conjunto de datos n-dimensional, en esta sección se describe el método para realizar un análisis de componentes principales.

Una vez que se tiene un conjunto de datos, el proceso de estandarización consiste en extraer la media y desviación estándar de cada variable (columna).

$$r_{i,j} = \frac{x_{i,j} - \bar{X}_j}{f_j}, \quad (5.5)$$

donde $r_{i,j}$ es el dato i de la variable j estandarizado, $x_{i,j}$ es el dato i de la columna j original y \bar{X}_j es el promedio del conjunto de datos de la variable j y f_j su desviación estándar.

Con los datos ya estandarizados se construye la matriz de correlaciones de acuerdo al método descrito anteriormente y se calculan los valores y vectores propios correspondientes. Cabe resaltar que si el conjunto de datos está representado por muchas variables el cálculo de los valores y vectores propios se convierte en una tarea extremadamente difícil de realizar a pie. Es por ello que se recurre a paqueterías como Mathematica o Python las cuales usualmente devuelven los vectores propios en orden de acuerdo al valor propio asociado,

²De hecho para un conjunto de datos n-dimensional es posible calcular $\frac{n!}{(n-2)! * 2}$ valores diferentes de covarianza

de mayor a menor. En PCA los valores propios miden que tanta es la varianza contenida en cada componente principal, por ende los valores propios mas grandes son mayores para las primeras componentes principales y disminuyen para las demás. Esto significa que las primeras componentes principales son aquellas que representan la mayor varianza del conjunto de datos (**figura 5.1**).

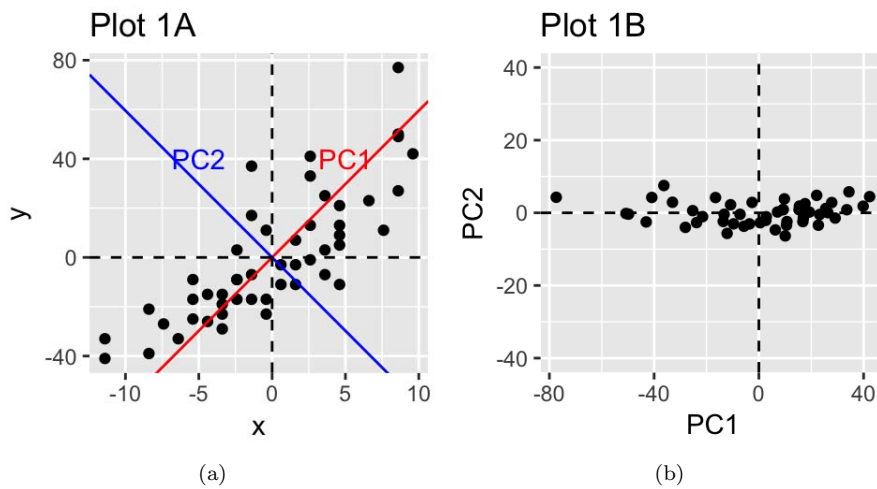


Figura 5.1: En a) se tienen unos datos representados en el sistema coordenado X-Y. Al identificar mediante ACP las direcciones con la mayor varianza es posible reducir la dimensión del sistema original y obtener b). Imagen tomada de [64]

Posteriormente es necesario escoger el número de componentes principales a conservar y esto puede llevarse a cabo de diferentes maneras. De acuerdo con Kaiser (1961) [65]:

- Un valor propio mayor que uno indica que la componente principal tiene más varianza que cualquiera de las variables originales. Este criterio es usualmente el utilizado para establecer un corte sobre las componentes principales a conservar y sólo es válido cuando el conjunto de datos está estandarizado.
- Es posible limitar el número de componentes principales a conservar de acuerdo con la varianza acumulativa explicada. Esto depende del porcentaje que uno desee conservar.

En el último caso puede uno ayudarse de un *Scree Plot* (figura 5.2) donde el número de componentes se escoge en el punto más allá del cual los demás valores propios son relativamente más pequeños ([66], [67]).

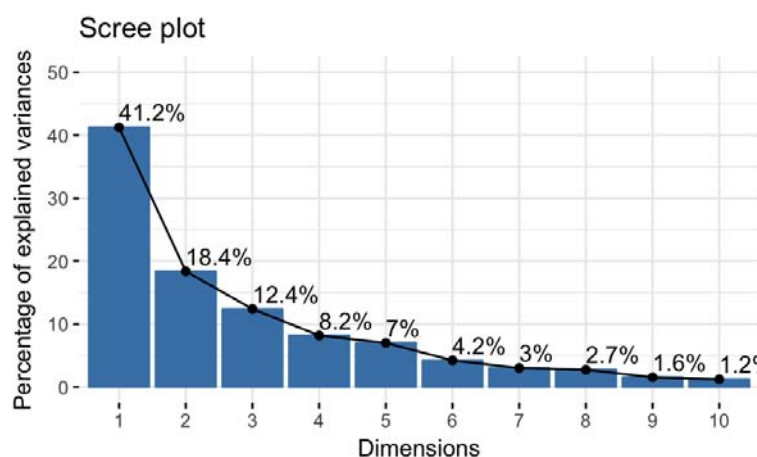


Figura 5.2: El *Scree Plot* ayuda a identificar la varianza acumulativa del conjunto de valores propios de la matriz de correlación. Tomada de [64]

Una vez seleccionadas las componentes y sus correspondientes vectores asociados, use usa la transpuesta de estos últimos para construir el vector de características: $C_{vect} = (eig_1^T, eig_2^T, eig_3^T)$. Como se puede ver, el vector de características está compuesto por los vectores propios asociados a las componentes principales.

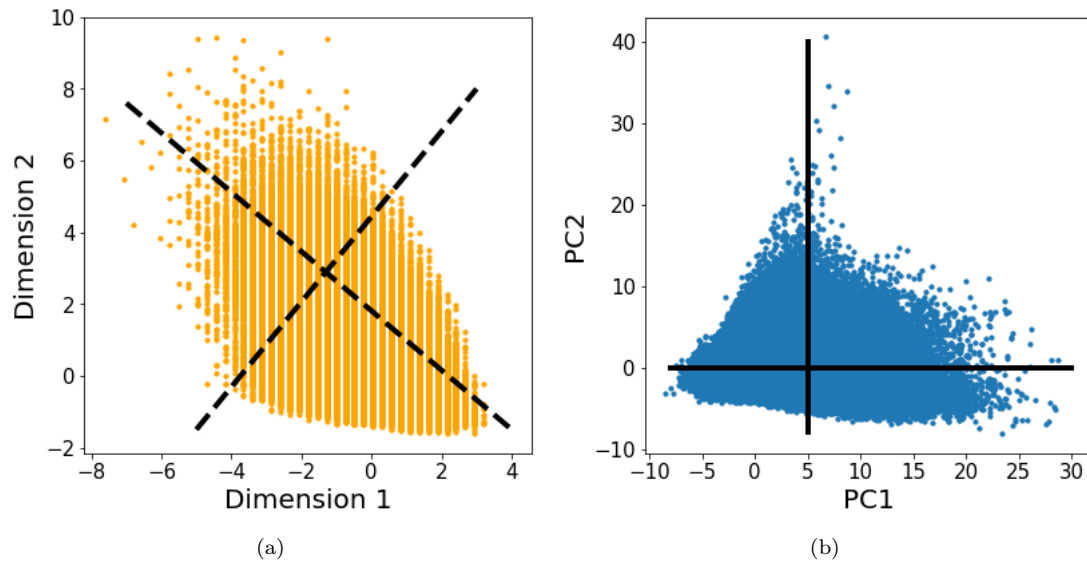


Figura 5.3: En el lado izquierdo se tienen dos dimensiones extraídas de los datos que fueron utilizados para esta tesis, mientras que en el lado derecho se han extraído las dos primeras componentes principales. Dado que el conjunto de datos es multidimensional no es sencillo ver la acumulación de los datos sobre las componentes principales como en el caso de 2 dimensiones.

Finalmente, el vector de características se utiliza para transformar los datos originales al nuevo espacio de componentes principales:

$$F = C_{vect} \times D^T, \quad (5.6)$$

donde C_{vect} es el vector de características y D^T el conjunto original de datos estandarizado y transpuesto. En el caso de dos dimensiones esta transformación corresponde a una rotación de los ejes originales a los ejes de las componentes principales (figura 5.1). Cuando se trabaja con datos multidimensionales es más difícil de ver de manera gráfica las componentes principales. La figura 5.3 muestra las primeras dos componentes principales extraídas del conjunto de datos que se utilizó para esta tesis. Puede verse (aunque de manera no tan clara) como se acumulan los datos sobre estas dos componentes principales.

En el siguiente capítulo se describe el análisis de componentes principales utilizado para transformar los datos originales provenientes de la reconstrucción de cascadas atmosféricas de HAWC y de esta forma reducir su dimensionalidad.

Capítulo 6

Deep Learning: Aprendizaje con redes neuronales

En los años recientes, el tema de la inteligencia artificial (IA) ha ido cobrando importancia. La gente habla de automóviles que se conducen solos, asistentes virtuales que programan actividades día a día entre otras cosas. Muchas de estas cosas podrían estar lejos de ocurrir aún y muchas otras podrían estar ya ocurriendo. En este capítulo se dará el contexto acerca del concepto de inteligencia artificial y lo que ésta incluye: *machine learning*¹ (ML) y *deep learning*² (DL).

6.1. Inteligencia Artificial

El concepto de IA se remonta a los años 50, momento en el que la ciencia computacional comenzaba a abrirse paso. Los pioneros de este campo se preguntaban hasta que punto se podía hacer que una computadora *pensara*. Las ramificaciones de esta cuestión aún se siguen explorando en la actualidad. Se podría dar una definición concisa del campo de las IA: *el esfuerzo para automatizar tareas intelectuales que normalmente son llevadas a cabo por humanos*[68]. Como tal el campo de la IA es muy general y abarca ML y DL pero también incluye otros acercamientos que no tienen nada que ver con aprendizaje alguno. Un ejemplo de esto fueron los primeros programas de ajedrez, que consistían en una serie de reglas fabricadas por los programadores y que como tal no clasificaban como ML. Durante mucho tiempo se pensó que la IA a nivel humano podría alcanzarse simplemente poniendo a los programadores a desarrollar una serie de reglas explícitas y complicadas para poder manipular el conocimiento. A este acercamiento se le conoce como *IA simbólica* y fue el paradigma dominante en IA desde los 1950s hasta finales de los 1980s (véase por ejemplo [69]).

Aunque la IA simbólica funcionaba de buena forma para resolver problemas lógicos bien definidos, como jugar ajedrez, pronto se convirtió en algo intratable a la hora de resolver problemas más complicados como clasificación de imágenes o traducciones de idiomas. En este punto la IA simbólica fue reemplazada por un nuevo campo: el DL.

6.2. *Machine Learning*

El concepto de ML surge de una pregunta muy interesante: ¿Podría una computadora *ir más allá de lo que sabemos ordenarle que haga* y aprender a realizar las tareas por sí misma? En lugar de que los programadores diseñen reglas para el procesamiento de datos ¿Podría una computadora aprender dichas reglas de manera automática con tan sólo ver los datos? Estas dos preguntas abren la puerta a un nuevo paradigma. En programación clásica, el paradigma de IA simbólica, el programador introduce las reglas en forma de programa y una serie de datos a ser analizados de acuerdo con dichas reglas para así obtener las respuestas. Con ML el programador introduce datos y las respuestas que se esperan de los datos y entonces recibe las reglas que pueden ser aplicadas a nuevos datos para producir respuestas originales.

En ML el sistema se entrena en lugar de programarse explícitamente. Al sistema se le presentan muchos ejemplos relevantes para una tarea y éste encuentra estructura estadística en dichos ejemplos, lo que le permite eventualmente hallar reglas para automatizar la tarea.

¹La traducción literal sería *aprendizaje máquina*

²La traducción literal sería *aprendizaje profundo*

Aunque el ML comenzó a florecer en los años 1990s, rápidamente se convirtió en el campo de la IA más popular y exitoso, llevado de la mano con la disponibilidad de hardware más rápido y de bases de datos cada vez más grandes. Otro punto a resaltar es que el ML está muy relacionado con la estadística y sin embargo difiere de ella de muchas y muy importantes maneras. Por ejemplo ML tiende a manejar enormes y complejos conjuntos de datos para los cuales el análisis estadístico clásico sería imposible de realizar. Como resultado, ML, y especialmente DL exhiben muy poca teoría matemática en comparación y están más orientados a la ingeniería.

6.3. Aprendiendo representaciones de los datos

Antes de definir DL y sus diferencias con otros algoritmos de aprendizaje, primero es necesario tener una idea de lo que hacen los algoritmos de ML. Un algoritmo de ML requiere de 3 cosas [68]:

- Datos de entrada. Si por ejemplo la tarea es sobre reconocimiento de discursos, los datos de entrada podrían ser archivos de sonido de personas hablando.
- Ejemplos de lo que se espera como salida. En una tarea de reconocimiento de discursos, estos podrían ser transcripciones de los archivos de sonido.
- Una forma de medir el desempeño del algoritmo. Esto es necesario para poder determinar que tanto difiere la salida del algoritmo de la salida esperada. Esta medida permite retroalimentar al algoritmo y mejorar su desempeño sobre la marcha. A este último paso de ajuste es a lo que se conoce como *aprendizaje*.

Mediante la exposición a ejemplos conocidos de entrada y de salida, un algoritmo de ML transforma sus datos de entrada en salidas útiles. El principal objetivo de ML y DL es transformar los datos en una forma que adquiera significado, es decir, aprender representaciones útiles de los datos de entrada; *representaciones*³ que se acerquen mucho a la salida que esperamos.

Todos los algoritmos de aprendizaje consisten en encontrar, de manera automática, transformaciones que conviertan a los datos en representaciones útiles para una tarea en específico. Dichas transformaciones pueden ser cambios de coordenadas, proyecciones lineales, traslaciones, etc. Los algoritmos de aprendizaje no encuentran estas operaciones por sí mismos, simplemente las buscan dentro de un conjunto predefinido de operaciones, al que se le conoce como *espacio de representaciones*.

Esto es lo que ML hace, buscar dentro de un espacio de posibilidades predefinido las representaciones útiles para un conjunto de datos de entrada.

6.4. Lo diferente de *Deep Learning*

DL es un campo muy específico dentro de ML: una nueva forma de aprender representaciones de los datos que tiene énfasis en aprender *sucesivas* capas, cada una con representaciones cada vez más significativas. DL no consiste en profundizar el aprendizaje, más bien se refiere a la idea de tener capas de representaciones sucesivas. La *profundidad* del modelo se refiere al número de capas que contribuyen a éste.

En este tipo de aprendizaje, las representaciones en forma de capas son aprendidas mediante modelos llamados *redes neuronales*⁴ estructuradas en capas, unas encima de otras.

Las figuras 6.1 y 6.2 muestran como se ve un algoritmo de DL y cómo es que las representaciones son aprendidas. En la figura 6.1 ejemplifica una red neuronal de profundidad dada que transforma una imagen de un dígito para reconocer de qué dígito de trata.

En la figura 6.2 se ve como la red transforma la imagen del dígito en representaciones que son cada vez más distintas de la imagen original, además de que son más informativas sobre el resultado final. puede pensarse en una red profunda como una operación de destilación de información, donde dicha información va pasando a través de filtros sucesivos y va saliendo cada vez más *pura* (es decir, más útil para realizar una tarea).

La información sobre cómo opera una capa con los datos de entrada se encuentra almacenada en los *pesos* de la capa, los cuales básicamente son un conjunto de números. La transformación que aplica cada

³En ML una representación es una forma de diferente de ver los datos – representar o codificar datos

⁴Aunque el término *red neuronal* es una referencia a la neurobiología y fue tomado utilizando la inspiración del entendimiento del cerebro, los modelos de DL no son modelos del cerebro.

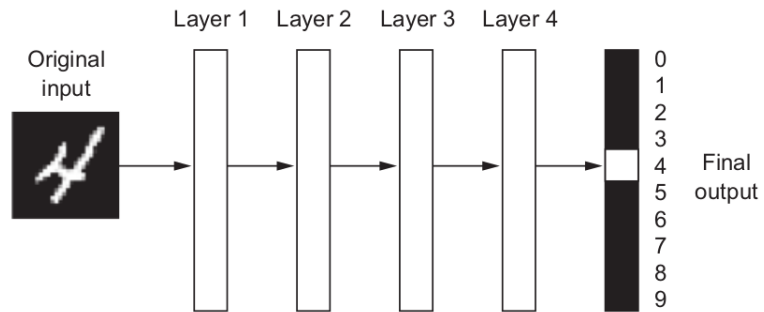


Figura 6.1: Ejemplo de una red neuronal profunda para clasificación de dígitos. Imagen tomada de [68].

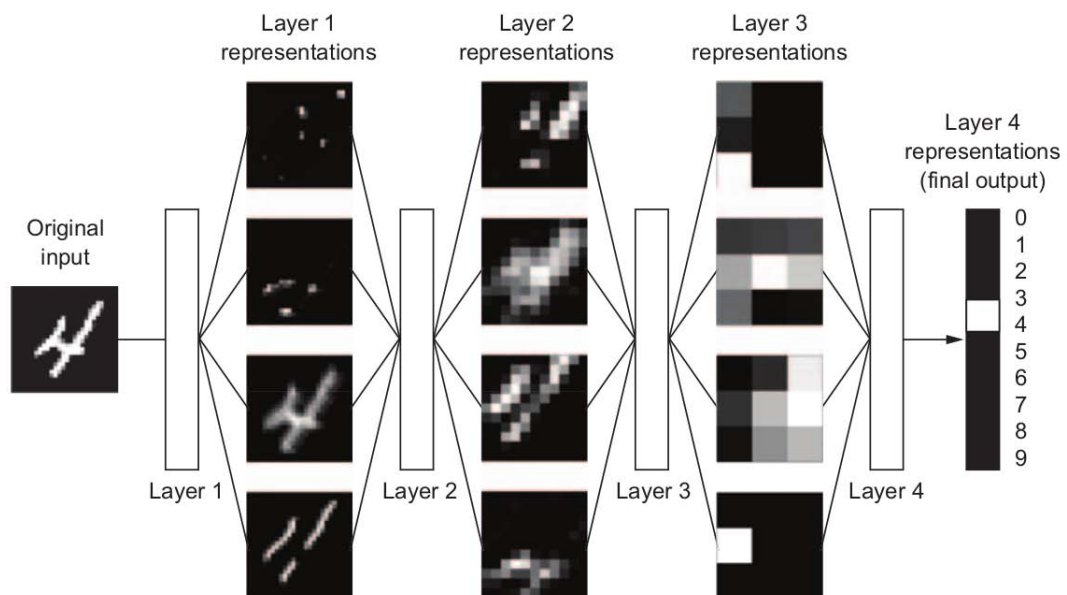


Figura 6.2: El modelo aprende representaciones de cada dígito. Imagen tomada de [68].

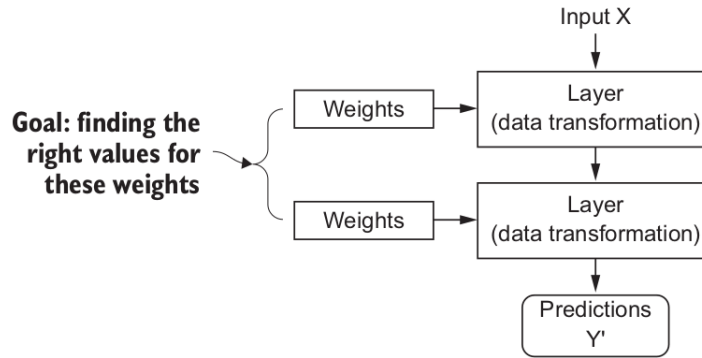


Figura 6.3: El objetivo de la red neuronal es hallar los valores correctos de los *pesos* con el objetivo de mapear de manera correcta los ejemplos con sus respectivos objetivos. Imagen tomada de [68].

capa a los datos está *parametrizada* por los pesos (figura 6.3). En este contexto, el *aprendizaje* implica hallar el conjunto de valores de los *pesos* de todas las capas de la red de tal forma que ésta pueda mapear correctamente los ejemplos de entrada y sus objetivos asociados.

Una red neuronal no funciona si su salida difiere de lo que uno espera obtener. Surge entonces la necesidad de poder medir qué tan lejos está la salida de la red con respecto a lo que se espera obtener. La *función de pérdida* (o *loss function*) se encarga de ésta tarea al comparar las predicciones de la red con el objetivo real (lo que se espera obtener) y medir qué tan alejados se encuentran entre sí. Usar ésta medida de distancia entre objetivos y predicciones como señal de retroalimentación para ajustar los valores de los pesos ayudará a reducir el valor de la función de pérdida mejorando la eficacia del aprendizaje. Ésta tarea se lleva a cabo mediante un *optimizador* que utiliza el algoritmo de propagación inversa [70] (del inglés *backpropagation algorithm*). En la práctica una red neuronal consiste de una serie de operaciones tensoriales encadenadas, cada una de las cuales tiene una derivada simple. Por ejemplo la red f compuesta de 3 operaciones tensoriales a , b , c cuyas matrices de peso son $W1$, $W2$, $W3$;

$$f(W1, W2, W3) = a(W1, b(W2, c(W3)))$$

El Cálculo dice que estas operaciones pueden derivarse mediante un método conocido como *regla de la cadena*: $f(g(x)) = f'(g(x)) * g'(x)$. Aplicar la regla de la cadena para el cálculo de los gradientes de la red neuronal es lo que hace el algoritmo de propagación inversa. Este algoritmo inicia con el valor de la pérdida final y trabaja desde las capas iniciales a las capas finales aplicando la regla de la cadena para calcular la contribución de cada parámetro en el valor de la pérdida.

6.5. Red Neuronal en acción

La manera de entender cómo funciona una red neuronal es aplicarla a ejemplos sencillos mientras se explica el procedimiento y las órdenes que se van ejecutando. Uno de los problemas más aplicados en ML es el problema de clasificación binaria, con el que se busca dividir cualquier conjunto de datos en dos categorías.

El ejemplo (tomado de [68]) con el que se trabajará esta sección tiene como objetivo aprender a clasificar críticas de películas como positivas o negativas de acuerdo con el texto contenido en cada crítica. El propósito final de este ejemplo es entender un algoritmo de clasificación binaria para después aplicarlo a los datos de HAWC y de esta manera aprender a diferenciar cascadas hadrónicas de electromagnéticas.

Se trabajó con la base de datos de películas de Internet (IMDb por sus siglas en inglés). Esta base de datos consiste en un conjunto de 50,000 críticas altamente polarizadas (muy negativas o muy positivas). Las críticas se dividen a la mitad en conjuntos de 25,000 críticas que servirán para el entrenamiento de la red mientras que el resto servirá para comprobación. Hay una subdivisión más en cada conjunto: el 50% de críticas son positivas y el otro 50% de críticas son negativas. La razón de la división inicial de las críticas es porque jamás se debe comprobar el modelo sobre los mismos datos con los que se entrenó. El hecho de que el modelo se comporte muy bien con los datos de entrenamiento no significa que se comportará igual de bien con datos que nunca ha visto. Es por ello que el conjunto de comprobación sirve para revisar el buen comportamiento de la red.

La paquetería utilizada lleva por nombre *Keras* [71], usa *Tensorflow* para las operaciones tensoriales y trabaja sobre Python.

Primero se importa el conjunto de datos de IMDb:

```
from keras.datasets import imdb
(train_data, train_labels), (test_data, test_labels) =
    imdb.load_data(num_words=10000)
```

Con *num_words=10000* se busca guardar solamente las primeras 10,000 palabras que aparecen con mayor frecuencia, descartando así las palabras raras. Luego, *train_data* y *test_data* son las listas de críticas; cada crítica es una lista de índices de palabras (que codifican una secuencia de palabras). Finalmente *train_labels* y *test_labels* son listas de 0's y 1's donde 0 es para críticas negativas y 1 para positivas.

Los datos en forma de listas no pueden utilizarse como entrada a la red neuronal; es necesario convertirlos a tensores (matrices):

```
1 import numpy as np
2 def vectorize_sequences(sequences, dimension=10000):
3     results = np.zeros((len(sequences), dimension))
4     for i, sequence in enumerate(sequences):
5         results[i, sequence] = 1.
6     return results
7 x_train = vectorize_sequences(train_data)
8 x_test = vectorize_sequences(test_data)
```

El código anterior funciona así:

Se crea una matriz cuya dimensión es $(\text{len}(\text{sequences}), \text{dimension})$. Posteriormente convierte algunos índices de resultados[i] a 1's. Se tienen al final los datos de entrenamiento y validación vectorizados. De la misma forma es posible vectorizar las etiquetas:

```
1 y_train = np.asarray(train_labels).astype('float32')
2 y_test = np.asarray(test_labels).astype('float32')
```

La forma más sencilla para entrenar la red es: vectores como datos de entrada y los etiquetas son escalares (0's y 1's). Para este tipo de problemas se utiliza un conjunto de capas del tipo *Dense* apiladas con activación *relu*: `Dense(16, activation='relu')`.

El argumento que se le pasa a cada capa *Dense* es el número de *unidades ocultas* de dicha capa. Una *unidad oculta* es una dimensión en el espacio de representación de la capa. Una capa *Dense* con activación *relu* implementa una serie de operaciones tensoriales dada como:

$$\text{output} = \text{relu}(\text{dot}(W, \text{input}) + b)$$

Tener 16 *unidades ocultas* implica que la matriz de peso, *W*, tendrá dimensión $(\text{input_dimension}, 16)$: el producto punto con *W* proyecta los datos de entrada a un espacio de representaciones 16-dimensional para posteriormente sumarle el vector *b* y aplicar la operación *relu*. La dimensionalidad del espacio de representaciones puede verse como *qué tanta libertad se le está dando a la red al momento de aprender representaciones*. Es claro que con más *unidades ocultas* la red puede aprender representaciones más complejas lo que también conlleva a un mayor costo computacionalmente hablando o a que la red aprenda patrones que no se espera aprender⁵.

A continuación se implementa la red neuronal:

```
1 from keras import models
2 from keras import layers
3 model=models.Sequential()
4 model.add(layers.Dense(16, activation='relu', input_shape=(10000,)))
5 model.add(layers.Dense(16, activation='relu'))
6 model.add(layers.Dense(1, activation='sigmoid'))
```

Se tienen dos capas intermedias con 16 unidades cada una (líneas 4 y 5) y una tercera capa (línea 6), la capa de salida, que sólo tiene una unidad pues su función es regresar una predicción de acuerdo con la crítica.

⁵Un ejemplo de esto es cuando la red tiene un gran desempeño con los datos de entrenamiento pero no con los datos de comprobación

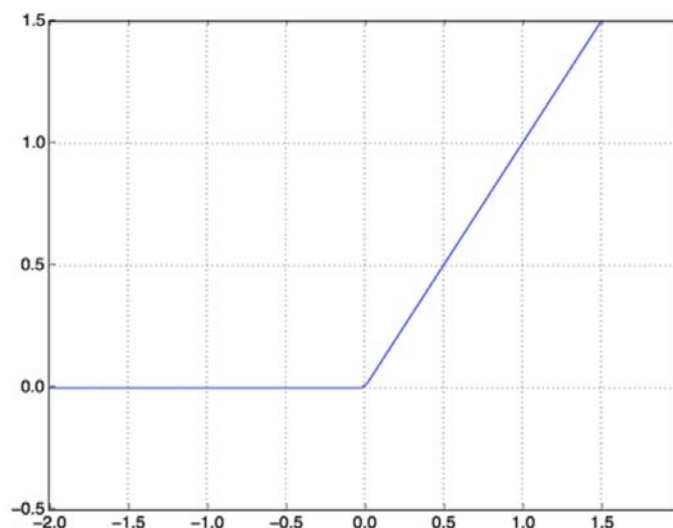


Figura 6.4: La función *relu* se encarga de hacer cero los valores negativos sin afectar a los positivos. Tomada de [68]

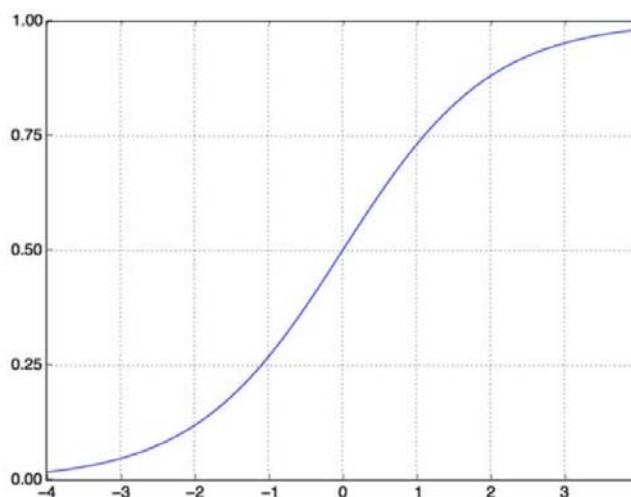


Figura 6.5: La función *sigmoid* acota valores arbitrarios al intervalo $[0,1]$. El resultado de esta operación puede interpretarse como una probabilidad. Tomada de [68].

Las capas intermedias utilizan *relu* (de rectified linear unit)⁶ como función de activación mientras que la capa de salida usa una activación *sigmoid* lo que regresa una probabilidad (entre 0 y 1) de que la muestra tenga etiqueta 1, es decir, que tan probable es que la crítica sea positiva.

Una función *relu* se encarga de hacer cero los valores negativos (figura 6.4) mientras que una función *sigmoid* acomoda valores arbitrarios dentro del intervalo $[0,1]$ (figura 6.5) devolviendo algo que podría interpretarse como una probabilidad.

El proceso de compilación se lleva con el siguiente comando:

```
1 model.compile(optimizer='rmsprop',
2               loss='binary_crossentropy',
3               metrics=['accuracy'])
```

En el momento de la compilación es necesario aplicar una función de pérdida y un optimizador. Para los problemas de clasificación binaria donde la salida del modelo es una probabilidad, se recomienda el uso de *binary_crossentropy* para la pérdida ya que es la más usada cuando se entrenan modelos que devuelven

⁶Esta función de activación le permite a la capa tener un espacio de representaciones más amplio al agregar operaciones no lineales.

probabilidades. Para monitorear la precisión del modelo que se entrena sobre datos que nunca ha visto, se toman 10000 críticas del conjunto original destinado a entrenamiento:

```
1 x_val=x_train[:10000]
2 partial_x_train=x_train[10000:]
3 y_val=y_train[:10000]
4 partial_y_train=y_train[10000:]
```

Se están tomando 10000 críticas con sus respectivas etiquetas (líneas 1 y 3) y se están dejando las restantes críticas y etiquetas para entrenamiento (líneas 2 y 4).

A continuación se entrena el modelo para 20 épocas (una época es una iteración sobre todas las muestras del entrenamiento) en grupos de 512 muestras. En cada época también se monitorea la precisión y pérdida del modelo en las 10000 muestras que se separaron del conjunto original.

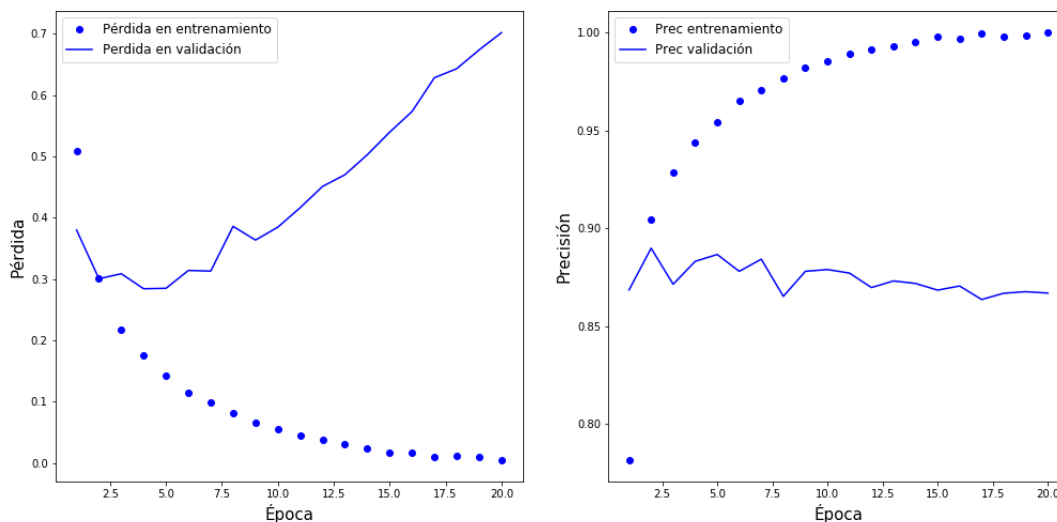
```
1 history=model.fit(partial_x_train ,
2                  partial_y_train ,
3                  epochs=20,
4                  batch_size=512,
5                  validation_data=(x_val , y_val))
```

Este ejemplo tarda alrededor de 20 segundos en terminar. Cuando termina de iterar en cada época hay una pausa para calcular la precisión y la pérdida en las 10000 muestras de validación.

El método `fit` regresa un objeto `History` que tiene diferentes atributos. Uno de ellos es `history`, un diccionario que contiene información sobre lo que ocurrió durante el entrenamiento. La manera de acceder a dicho diccionario se describe a continuación:

```
1 >>> history_dict = history.history
2 >>> history_dict.keys()
3 [u'acc', u'loss', u'val_acc', u'val_loss']
```

Como puede verse (línea 3), el diccionario `history_dict` contiene 4 elementos, los primeros dos indican la precisión y pérdida en el entrenamiento y los últimos dos lo mismo pero en la validación. Para saber que es lo que está ocurriendo en cada época de entrenamiento y validación se crea una gráfica de la pérdida (figura 6.6(a)) y una de la precisión (figura 6.6(b)).



(a) Pérdida en entrenamiento y validación

(b) Precisión en entrenamiento y validación

Figura 6.6: Pérdida y precisión en entrenamiento y validación

Como es de esperarse, en entrenamiento la pérdida decrece con cada época mientras que la precisión se incrementa de la misma forma. El problema surge porque tanto la precisión como la pérdida parecen alcanzar un extremo en la validación alrededor de la cuarta época. A partir de ahí la pérdida se incrementa y la precisión disminuye con cada época. Lo que se está viendo en este caso es el claro ejemplo de que el modelo se está especializando demasiado en los datos de entrenamiento, de tal forma que cuando se le presentan nuevos datos no sabe cómo clasificarlos. A este fenómeno se le conoce como sobreajuste (u *overfitting*).

En este caso es posible librarse del sobreajuste entrenando nuevamente la red pero solamente para 4 épocas:

```

1 model = models.Sequential()
2 model.add(layers.Dense(16, activation='relu', input_shape=(10000,)))
3 model.add(layers.Dense(16, activation='relu'))
4 model.add(layers.Dense(1, activation='sigmoid'))
5 model.compile(optimizer='rmsprop',
6 loss='binary_crossentropy',
7 metrics=['accuracy'])
8 model.fit(x_train, y_train, epochs=4, batch_size=512)
9 results = model.evaluate(x_test, y_test)

```

El resultado obtenido para este nuevo entrenamiento es:

```

1 >>> results
2 [0.2929924130630493, 0.8832799999999999]

```

Como puede verse, con 4 épocas de entrenamiento se elimina el sobreajuste y se obtiene una precisión del 88% sobre los datos de validación. Cabe aclarar que es posible mejorar la precisión si se cambian, por ejemplo, las unidades o el número de capas teniendo siempre en cuenta el sobreajuste.

Una vez entrenada y validada, la red puede usarse para hacer predicciones. Retomando el conjunto de comprobación (que consta de 25000 críticas) y usando el método `predict`:

```

1 >>> model.predict(x_test)
2 array([[ 0.98006207]
3 [ 0.99758697]
4 [ 0.99975556]
5 ...,
6 [ 0.82167041]
7 [ 0.02885115]
8 [ 0.65371346]], dtype=float32)

```

Nótese que la red puede clasificar con certeza algunas muestras (0.99 o más o 0.01 o menos) mientras que para algunas críticas no queda muy claro a que conjunto pertenecen (0.6,0.4).

Como comentarios finales a esta sección pueden decirse algunas cosas:

- No hay un manual que te diga el número de capas o de unidades a usar en una red.
- Siempre hay que cuidar el sobreajuste ya que es lo que indica cuando la red se está especializando demasiado en el entrenamiento.
- Dependiendo del tipo de datos a clasificar se puede tener un mejor o peor desempeño de la red.
- Es importante dar a la red datos que brinden información que permita diferenciar una categoría de otra en el caso de una clasificación binaria.

Una vez que se han abarcado todos los conceptos necesarios, en el siguiente capítulo se procede con el análisis y resultados obtenidos.

Capítulo 7

Análisis y resultados

El objetivo principal de este trabajo es mejorar la manera en que se discriminan cascadas hadrónicas de cascadas electromagnéticas a bajas energías (unos cientos de GeV hasta máximo 1 TeV). La forma en que esto se llevó a cabo fue mediante el desarrollo e implementación de un algoritmo de aprendizaje que se entrenó utilizando datos de simulación. Ya que los algoritmos para clasificar hadrones y fotones funcionan bien para energías mayores a 5 TeV, el análisis se centró en seleccionar de la simulación las cascadas cuya energía reconstruida se encuentra entre 100 GeV y 1000 GeV. En este estudio se utilizó la energía estimada con *Neural Net* (sección 3.4.2).

Para este trabajo se usaron todas las simulaciones disponibles (alrededor de 16 archivos de simulación), las cuales se encuentran divididas en declinaciones, desde -35 hasta 75 grados. La simulación, con los cortes de selección ya aplicados, consta de alrededor de 12 millones de eventos, de los cuales solamente se tomaron 700,000 eventos (alrededor de 30,000 en cada declinación mitad fotones y mitad protones) seleccionados de manera aleatoria entre todos los archivos de simulación.

La diferencia con otros algoritmos de aprendizaje es que en este trabajo se aplica un análisis de componentes principales a 39 variables contenidas originalmente en los archivos de simulación. De este procedimiento se seleccionaron 9 componentes que explican de manera acumulativa el 68.69% de la varianza total del conjunto de datos. La sección 7.1 describe la selección de variables en los archivos, tanto de simulación como de datos reales. En la sección 7.1.1 se explica la aplicación del análisis de componentes principales, en la sección 7.1.2 se explica el entrenamiento de la red, en la sección 7.2 se describe la comprobación¹ con datos de la Nebulosa del Cangrejo y en la sección 7.3 se busca el GRB 170817A.

7.1. Preparación de la simulación

En la sección 3.5 se habló acerca de las variables contenidas en los archivos de datos y de simulación. Las variables utilizadas se encuentran en las tablas 3.4 y 3.5 y son un total de 39 variables. El total de eventos de las simulaciones ya seleccionadas es de 12 millones. Se tomó una muestra aleatoria con 700,000 eventos (350,000 fotones y 350,000 protones) para entrenar la red neuronal. A este conjunto de datos se le aplicó un análisis de componentes principales, ACP, que se explica a continuación. La figura 7.1 muestra un ejemplo de la concordancia entre datos de simulación y datos reales para un bin de análisis de HAWC.

7.1.1. Componentes principales

El ACP requiere que los datos estén estandarizados, es decir, que cada variable tenga varianza cero y desviación estándar 1. Con ACP se logró reducir el conjunto original de datos conservando así 9 variables, las que contienen la máxima varianza posible. La tabla 7.1 muestra las componentes principales obtenidas con la varianza individual y acumulada de cada una, además del valor propio asociado.

La figura 7.2 muestra el *Scree Plot* con las componentes principales y el porcentaje de varianza tanto individual como acumulativa explicado.

¹En la sección 6.5 se explica el porqué del uso de un conjunto de comprobación y uno de validación.

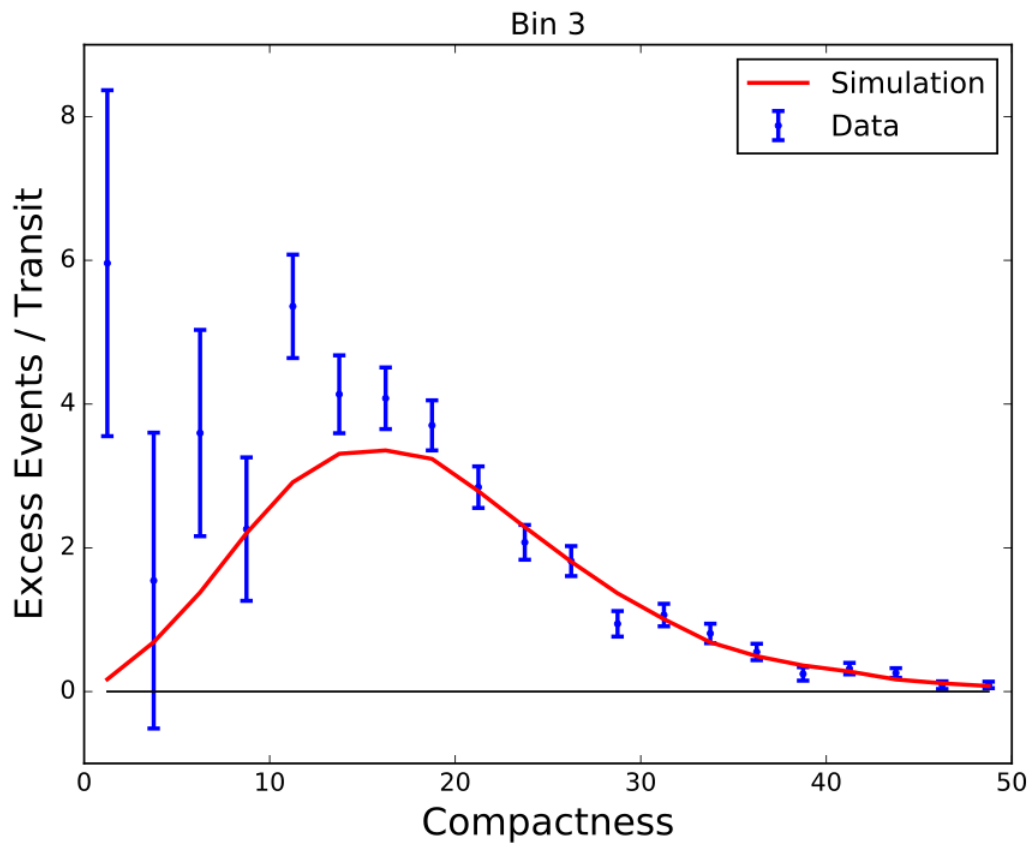


Figura 7.1: Concordancia entre la simulación y datos reales para un bin de análisis estándar de HAWC. Lo que se observa es que el número de eventos observado y simulado para la Nebulosa del Cangrejo están en concordancia. Imagen tomada de [57].

De acuerdo al procedimiento descrito en el capítulo de ACP, una vez escogidas las 9 componentes principales, se toman sus vectores propios asociados para construir la matriz de características con la cual se transforman los datos iniciales que estaban compuestos por 39 variables. La nueva matriz tiene dimensión $700,000 \times 9$ (700,000 eventos y 9 componentes principales)

7.1.2. Entrenamiento de la red

La red neuronal que se implementó para este estudio es una modificación de la red de ejemplo vista en la sección 6.5. Lo primero fue entrenar una red que sobreajustara para después comenzar a modificar los parámetros (capas y número de unidades por capa principalmente) y así desaparecer el sobreajuste. Es importante hacer notar que así como existe el sobreajuste existe también el inverso, cuando la red no ha aprendido representaciones importantes a partir de los datos y los resultados de entrenamiento y validación están correlacionados.

Al inicio se mencionó que el conjunto total de entrenamiento constaba de 700,000 eventos. Este conjunto se dividió en 2 partes de 350,000 eventos cada una. La primera mitad se utilizó para el entrenamiento y la segunda mitad para comprobación. Los 350,000 eventos de entrenamiento a su vez se dividieron en dos conjuntos de 150,000 y 200,000 eventos. La red entonces se entrenó con 200,000 eventos, se usaron los restantes 150,000 para validación, es decir, este conjunto sirve para revisar que la red tiene un buen desempeño y no sobreajusta.

La configuración de la red que produce un sobreajuste puede verse a continuación:

```
model = models.Sequential()
model.add(layers.Dense(32, activation='relu', input_shape=(9,)))
model.add(layers.Dense(32, activation='relu'))
model.add(layers.Dense(16, activation='relu'))
model.add(layers.Dense(16, activation='relu'))
model.add(layers.Dense(16, activation='relu'))
```

Componente	Valor propio	Porcentaje de varianza individual	Porcentaje de varianza acumulada
1	11.5089675	29.5101733	29.5101733
2	4.4754806	11.4755913	40.9857646
3	3.5582864	9.1238114	50.109576
4	1.6267218	4.1710817	54.2806577
5	1.3554206	3.4754375	57.7560952
6	1.1966677	3.0683787	60.8244739
7	1.0361549	2.6568074	63.4812813
8	1.0109387	2.5921505	66.0734318
9	1.0039186	2.5741503	68.6475821
10	0.9902357	2.5390660	71.1866481
11	0.9710702	2.4899236	73.6765717
12	0.9599368	2.4613765	76.1379482
13	0.9451972	2.4235826	78.5615308
...
39	0.0026150	0.0067052	100

Tabla 7.1: Componentes principales obtenidas del ACP, los valores están ordenados por varianza individual de mayor a menor. La fila en color verde indica el límite para el cual se dejan de considerar las posteriores componentes por no contribuir significativamente a la varianza total acumulada.

```

model.add(layers.Dense(16, activation='relu'))
model.add(layers.Dense(16, activation='relu'))

model.add(layers.Dense(1, activation='sigmoid'))

model.compile(optimizer='rmsprop',
              loss='binary_crossentropy',
              metrics=['binary_accuracy'])

"""Entrenamiento y validacion"""
history = model.fit(train_data, train_labels,
                   epochs=150,
                   batch_size=512,
                   validation_data=(val_data, val_labels))

history_dict = history.history
history_dict.keys()

```

Los resultados pueden verse en las figuras 7.3(a) y 7.3(b).

Una de las maneras para reducir el sobreajuste es reduciendo el poder de la red, ya sea reduciendo las capas, las épocas o el número de unidades. Las figuras 7.4(a) y 7.4(b) muestran nuevamente la red pero ahora reduciendo el número de unidades en las dos primeras capas, de 32 a 16 y manteniendo los demás parámetros sin modificar.

Tomando en cuenta que con una red menos poderosa se logró retrasar el sobreajuste, se consideró que 60 épocas son adecuadas y a partir de ahí se modificaron los demás parámetros de la red sin conseguir una mejora mayor a 76 % pero consiguiendo que no haya sobreajuste.

En este punto se entrenó nuevamente la red, con los parámetros finales y se utilizó el conjunto de comprobación para validar el entrenamiento. A continuación la configuración final de la red, la cual arrojó los resultados en pérdida y precisión de las figuras 7.5(a) y 7.5(b).

```

train_data=train[0:350000,:]
test_data=train[350000:,:]
train_labels=labels[0:350000]
test_labels=labels[350000:]

model = models.Sequential()
model.add(layers.Dense(16, activation='relu', input_shape=(9,)))

```

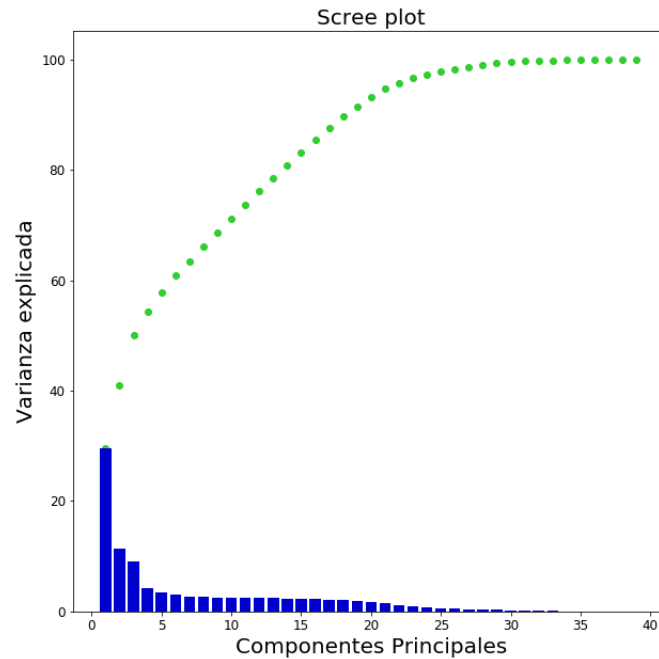



Figura 7.2: Con el *Scree Plot* se pudieron seleccionar las componentes a considerar para entrenar la red.

```

model.add(layers.Dense(16, activation='relu'))
model.add(layers.Dense(16, activation='relu'))
model.add(layers.Dense(16, activation='relu'))
model.add(layers.Dense(1, activation='sigmoid'))

model.compile(optimizer='rmsprop',
              loss='binary_crossentropy',
              metrics=['binary_accuracy'])
model.fit(train_data, train_labels,
          epochs=60,
          batch_size=512)
results=model.evaluate(test_data, test_labels)

print("loss, accuracy", results)
#save model
# serialize model to JSON
model_json = model.to_json()
with open("model_G_H_alldec_4lyr_16un.json", "w") as json_file:
    json_file.write(model_json)
    # serialize weights to HDF5
model.save_weights("model_G_H_alldec_4lyr_16un.h5")
print("Se guardo el modelo")

```

La red anterior se guardó para su posterior utilización sobre datos reales adquiridos con HAWC.

7.2. Comprobación de la red

Para probar la validez de la red se utilizó un objeto muy brillante, conocido y utilizado en HAWC: La Nebulosa del Cangrejo, un remanente de supernova y la fuente estática más brillante en el cielo norte por arriba de 1 TeV.

A continuación se describe la preparación de los datos para su análisis con la red y la creación de los mapas.

Para crear la respuesta del detector, necesaria para los mapas de significancia, se requieren los archivos

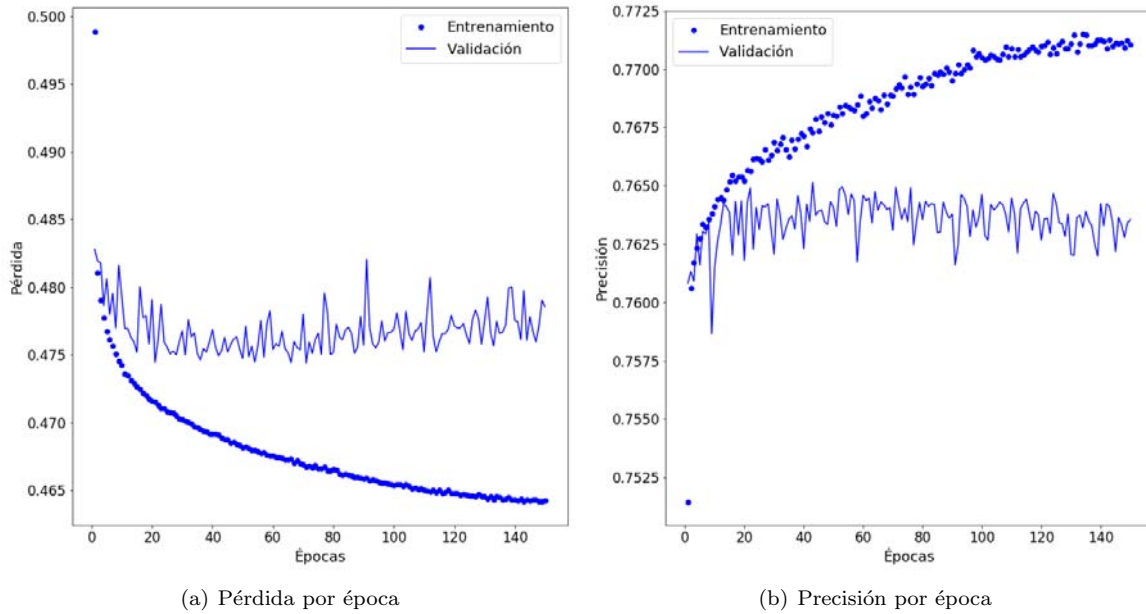


Figura 7.3: Pérdida y precisión en entrenamiento y validación con respecto a la época (iteración sobre el conjunto de datos). El sobreajuste comienza a partir de la época 40, donde la pérdida y la precisión en validación comienzan a empeorar con respecto al entrenamiento.

de simulación con la predicción de la red neuronal. Entonces se aplicó la red neuronal a los 12 millones de eventos contenidos en los archivos de simulación. Este procedimiento genera un archivo de predicciones, las cuales hay que agregar a los archivos de simulación. Dadas las limitaciones del tipo de archivo (xcd)² fue necesario crear nuevos archivos de simulación. Estos nuevos archivos de simulación son idénticos a los antiguos y la única diferencia reside en que los nuevos archivos ya incluyen la predicción de la red neuronal.

7.2.1. El *Crab strip*

Los datos de la Nebulosa del Cangrejo consisten en una serie de archivos seleccionados de tal forma que contienen tránsitos de ésta fuente en el campo de visión de HAWC, cada tránsito de la Nebulosa dura alrededor de 5 horas. Se tomaron alrededor de 5,400 archivos de este tipo, correspondientes a 189 horas de datos. Se aplicó la selección de eventos con energía entre 100 GeV y 1 TeV (rango en el que se entrenó la red), se transformaron los datos a través del vector de características de las componentes principales y se aplicó la red neuronal obteniendo archivos de predicciones. De la misma forma que con la simulación se crearon nuevamente los archivos de datos agregando la predicción de la red. La distribución de predicciones para uno de los 5400 archivos de datos puede verse en la figura 7.6. Cabe mencionar que se utilizaron las mismas variables que se usaron para la simulación, de otro modo no se garantiza que el desempeño de la red sea satisfactorio al momento de hacer predicciones.

Para la creación de mapas se define un archivo de cortes de selección, estos cortes definen los bins de análisis. En cada paso del análisis se utilizan las definiciones de los bins. A continuación los dos tipos de cortes, el corte del bin 0 oficial y el nuevo bin 0 que contiene la predicción de la red neuronal.

```
0 "(rec.nChTot > 800) && (rec.nChAvail > 0.9*rec.nChTot)
&& (rec.nHit >= 0.044*rec.nChAvail)
&& (rec.nHit < 0.067*rec.nChAvail)"

0NN "(rec.nChTot > 800) && (rec.nChAvail > 0.9*rec.nChTot)
&& (rec.nHit >= 0.044*rec.nChAvail)
&& (rec.nHit < 0.067*rec.nChAvail) && (rec.NNPred >= X)"
```

²El tipo de archivo xcd es muy poderoso pero tiene como desventaja que si se quiere agregar una nueva variable a un archivo .xcd se tiene que generar un nuevo archivo .xcd que incluya todas las variables del antiguo archivo más la nueva variable. Este nuevo archivo se llena evento a evento

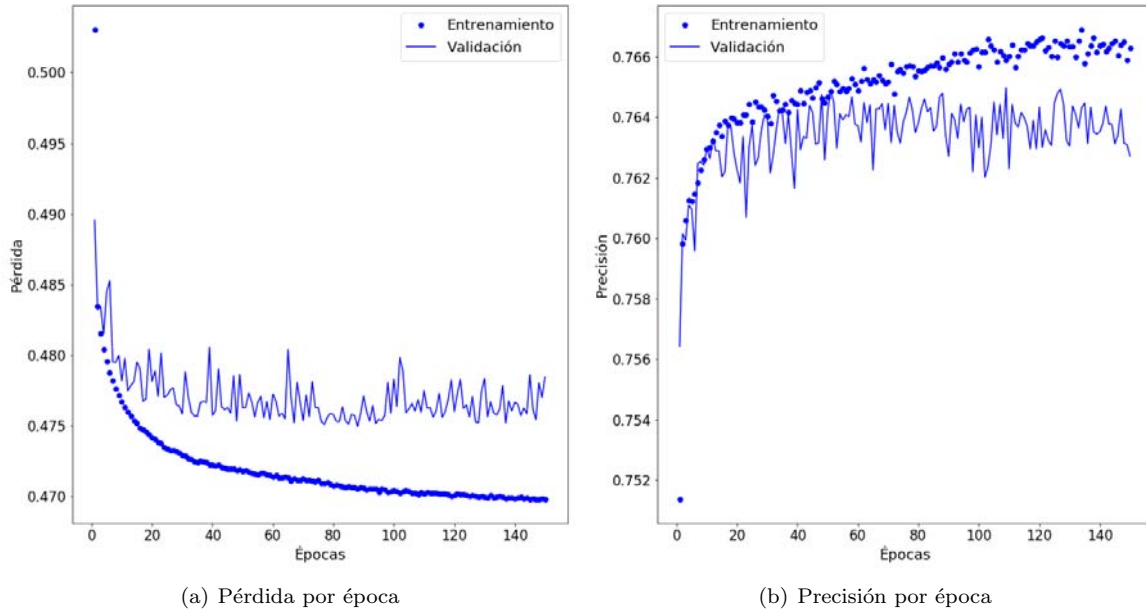


Figura 7.4: Pérdida y precisión en entrenamiento y validación con respecto a la época para una red con menos unides. El sobreajuste aparece a partir de la época 60.

Los cortes que se están definiendo van sobre el número de PMT's que están funcionando al momento de guardar los datos correspondientes a las cascadas y al número de PMT's que se activan con respecto al número de PMT's en funcionamiento. Para propósitos de calidad y consistencia, los cortes originales no se movieron y simplemente se agregó el corte a la nueva variable creada con la red neuronal (rec.NNpred). Para saber cual sería el corte de rec.NNpred óptimo, se hicieron mapas de significancia para diferentes valores de X . La figura 7.7 muestra como va la significancia de la Nebulosa del Cangrejo para los diferentes valores de X . La figura además muestra el mapa de significancias suponiendo fuentes con extensión de 0.5 y 1 grado.

La figura 7.8 muestra el mapa de significancias de la Nebulosa del Cangrejo con el corte oficial y con el valor $\text{rec.NNpred} \geq 0.4$.

La figura 7.8(b) muestra el mapa con los cortes oficiales mientras que la figura 7.8(a) utiliza el corte a rec.NNpred ; la tabla 7.2 muestra los tipos de cortes con la significancia máxima y ubicación obtenida para cada mapa así como la ubicación de la Nebulosa dada por el catálogo de HAWC [72].

Corte	Significancia máxima	Ubicación (A.R.,Dec)
Oficial	2.54	84.64, 22.51
Oficial con $\text{rec.NNpred} \geq 0.4$	3.65	84.73, 22.10
Ubicación real		83.63, 22.02

Tabla 7.2: Valores de significancia obtenidos para los diferentes tipos de cortes aplicados. El corte $\text{rec.NNpred} \geq 0.4$ da la mayor significancia de la Nebulosa del Cangrejo.

Como puede verse, la mayor significancia se obtiene utilizando el corte $\text{rec.NNpred} \geq 0.4$. Esto muestra un aumento del 43.7% en la significancia con respecto al corte oficial. La búsqueda del GRB 170817A se realizó tomando en cuenta el corte con el que se obtiene una mayor significancia de la Nebulosa del Cangrejo. La siguiente sección muestra el proceso de búsqueda.

7.3. Búsqueda del GRB 170817A

La zona ubicada por LIGO-Virgo entró al campo de visión de HAWC aproximadamente 8 horas después de que fue detectado el GRB. Mediante un GCN[73] se reportó esa búsqueda con HAWC para un índice espectral de -2.5 no encontrando nada significativo.

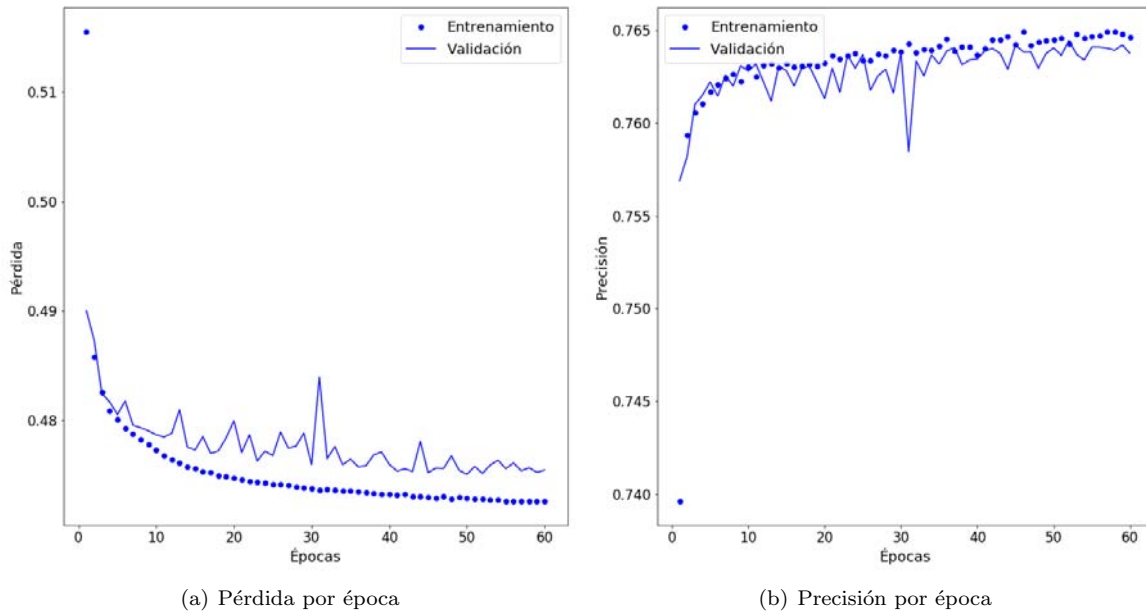


Figura 7.5: Pérdida y precisión en entrenamiento y validación con respecto a la época para la configuración final de la red.

Los datos tomados para este análisis abarcan de las 20:55:09 UTC a las 22:55:09 UTC del 17 de Agosto de 2017, alrededor de 2 horas. La creación de los mapas se llevó a cabo de la misma manera que con el análisis del *Crab strip*; se seleccionaron los eventos con energías entre 100 GeV y 1 TeV, se transformaron con la matriz de características del ACP y se les aplicó la red neuronal. La figura 7.6 muestra el histograma de las predicciones aplicadas a un conjunto de archivos de datos.

Los mapas de significancia con los dos tipos de cortes se muestran en las figuras 7.9(b) y 7.9(a).

Como se observa en la figura, nada significativo aparece en la región reportada por el GCN de HAWC, el punto de máxima significancia se encuentra ubicado en A.R. 178.33, Dec -21.86 con un valor de 3.69σ . Esta región se encuentra muy lejos de la ubicación reportada del *trigger* de LIGO y corresponde a algunos píxeles en el mapa solamente. Con los cortes oficiales la región de máxima significancia está ubicada en A.R. 180.97, Dec -18.05 con un valor de 2.15σ .

No habiendo encontrado nada significativo en la región cercana al *trigger* de LIGO se procedió a estudiar todo el campo de visión de HAWC para buscar otras regiones significativas.

El mapa del campo de visión de HAWC se encuentra en la figura 7.10. Al mapa se le superpusieron las fuentes del catálogo de TeVCat para revisar alguna asociación por ubicación con las regiones más significativas encontradas.

Una de las regiones de alta significancia del mapa se corresponde con la ubicación de una de las fuentes de TevCat, el blazar Markarian 421. La figura 7.11 muestra el zoom aplicado a la región de Mrk421 (el cual se encuentra en A.R. 166.11 deg, Dec 38.16 deg [72]).

La máxima significancia de la región mostrada anteriormente es 3.96 y se encuentra ubicada en A.R. 165.32 deg, Dec 38.49 deg, apenas desplazada un grado con respecto a la ubicación dada por el catálogo de HAWC.

A continuación se superpuso el catálogo de Fermi [74], figura 7.12 En este caso no se encontró asociación alguna de fuentes de Fermi con regiones de alta significancia de HAWC.

A pesar de que el objetivo principal de esta tesis no se logró, si se logró mejorar la separación de cascadas hadrónicas y electromagnéticas a bajas energías (entre 100 GeV y 1 TeV) con respecto al análisis oficial utilizado en HAWC para bajas energías.

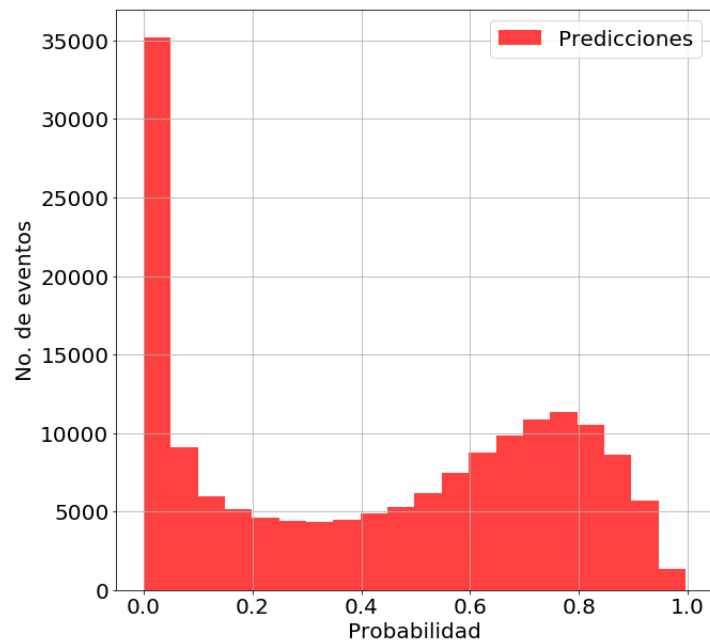


Figura 7.6: El histograma muestra cómo se ven los eventos cuando han sido clasificados por la red neuronal. En el eje x se muestra la probabilidad de cada evento de ser un fotón; probabilidades cercanas a 1 indican que el evento en cuestión corresponde a una cascada electromagnética mientras que probabilidades cercanas a 0 corresponden a que el evento corresponde a una cascada hadrónica. Es de esperarse que la mayoría de los eventos sean correspondientes a cascadas hadrónicas ya que es lo que más se detecta. Este histograma corresponde a predicciones realizadas con datos reales tomados con HAWC

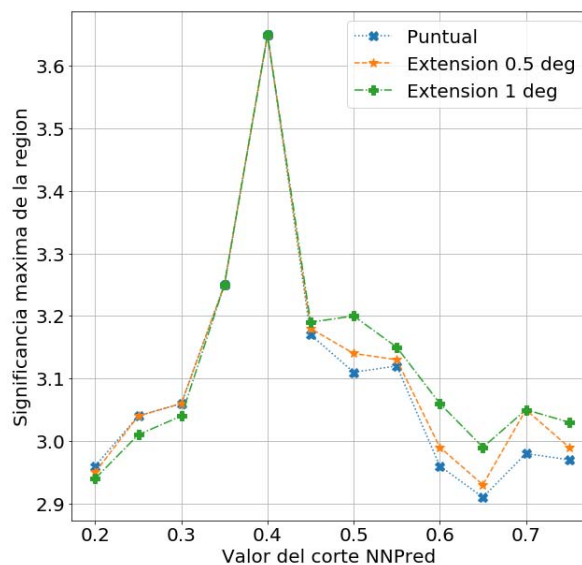


Figura 7.7: Significancia de la Nebulosa del Cangrejo con respecto al valor de rec.NNPred. Como puede verse, la mayor significancia se obtiene con el valor $\text{rec.NNPred} \geq 0.4$.

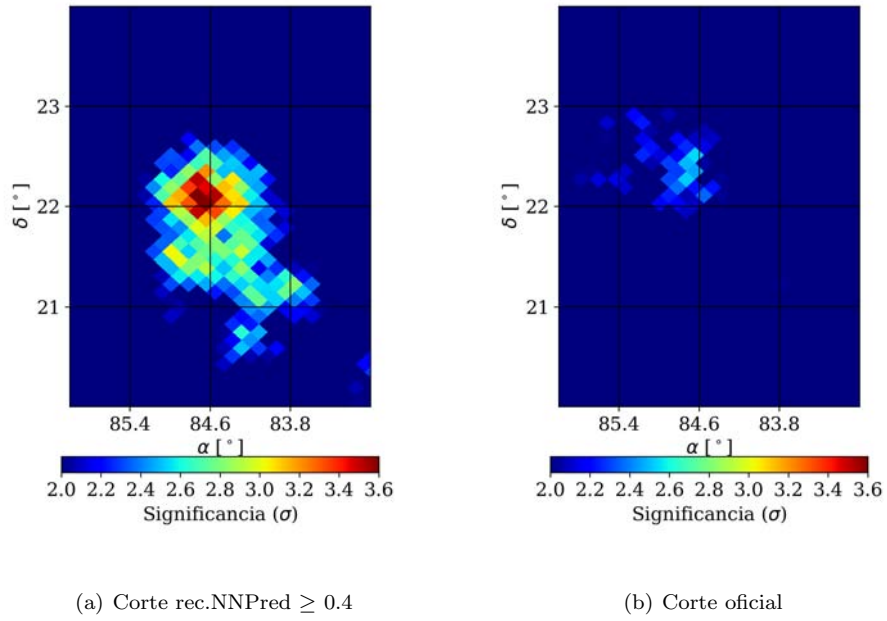


Figura 7.8: Mapas de significancia de la Nebulosa del Cangrejo con ambos tipos de cortes.

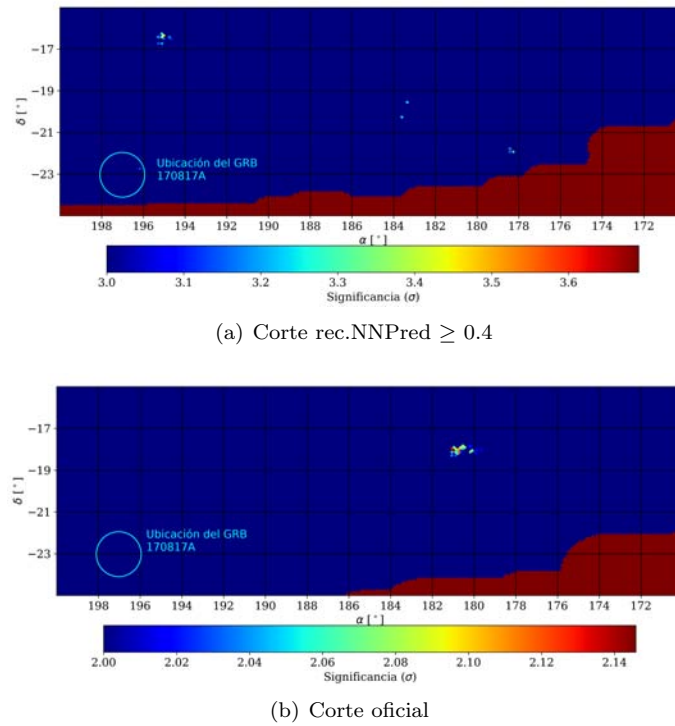


Figura 7.9: Mapas de significancia en una región alrededor de la ubicación del GRB 170817A. Las regiones rojizas son un efecto de falta de estadística. La región donde HAWC reportó el límite superior se encuentra encerrada en un círculo

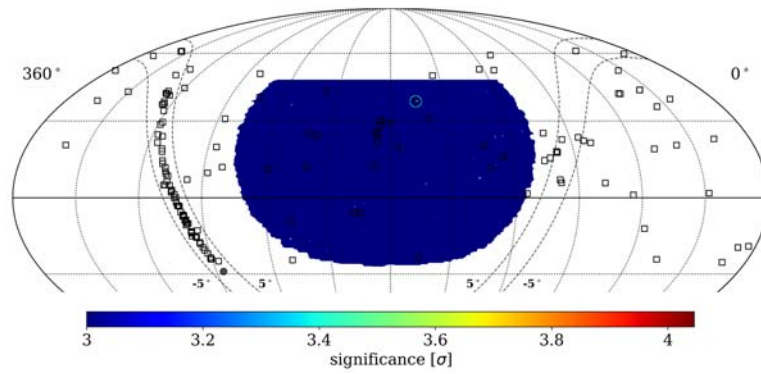


Figura 7.10: Mapa del campo de visión de HAWC con las regiones de mayor significancia. Se han superpuesto las fuentes del catálogo TeVCat. El círculo de color cian indica donde hay una correspondencia en posición entre la fuente de TeVCat y la región de más alta significancia en el mapa.

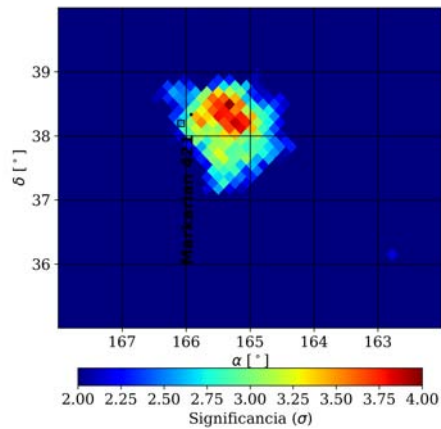


Figura 7.11: Mapa de significancias alrededor de Mrk421 para los días 15 a 20 de Agosto de 2017. Como puede verse, la fuente de TeVCat Mrk421 está muy cerca de esta región de alta significancia.

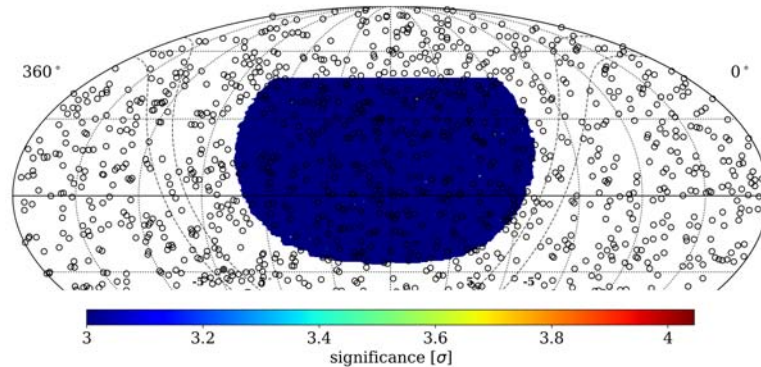


Figura 7.12: Mapa del campo de visión de HAWC en el que se superpusieron las fuentes del catálogo 3FGL de Fermi.

Capítulo 8

Conclusiones y trabajo a futuro

Con el fin de mejorar la manera en que se discriminan cascadas hadrónicas de cascadas electromagnéticas a bajas energías de HAWC (100 GeV-1 TeV), se desarrolló un algoritmo de aprendizaje con redes neuronales. Dicho algoritmo fue entrenado con datos de las simulaciones disponibles en la colaboración HAWC los cuales constan de 16 archivos separados por declinaciones en los que se incluyen eventos correspondientes a cascadas atmosféricas reconstruidas. La red neuronal se entrenó con 700,000 eventos de cascadas hadrónicas y electromagnéticas, seleccionados de manera aleatoria de entre todos los archivos de simulación.

Previo al proceso de entrenamiento se realizó un análisis de componentes principales a los eventos con el fin de reducir la dimensionalidad del conjunto de datos original (de 39 variables a 9 componentes principales). Estas 9 componentes principales explican el 68.64% de la varianza total contenida en las 39 variables seleccionadas originalmente. La red neuronal entrenada sobre los 9 parámetros (componentes principales) alcanzó una precisión de 76% en la clasificación de datos.

Posteriormente se aplicó la red neuronal sobre todos los archivos de simulación con el fin de crear archivos adecuados para la generación de respuestas del detector que servirían para los análisis de mapas posteriores.

Con el fin de comparar la eficacia de la red neuronal con el análisis a bajas energías de HAWC, se utilizaron 189 horas de datos correspondientes a tránsitos de la Nebulosa del Cangrejo (alrededor de 36 tránsitos). A estos archivos se les aplicó la red neuronal y se generaron nuevos archivos (xcd) donde se incluye la predicción de la red, la cual se definió con el nombre *rec.NNPred*.

Una vez que se obtuvieron los nuevos archivos xcd, se procedió a la creación de mapas de significancia. Se hicieron mapas considerando diferentes valores de límite en la variable *rec.NNPred* (desde $\text{rec.NNPred} \geq 0.2$ hasta $\text{rec.NNPred} \geq 0.75$) con el fin de encontrar el valor óptimo que devolviera la mayor significancia de la Nebulosa del Cangrejo. Se encontró que $\text{rec.NNPred} \geq 0.4$ devuelve el máximo valor de significancia (3.65σ) y que éste valor es 43% más alto que el del corte oficial a bajas energías (2.54σ).

El nuevo algoritmo se utilizó entonces para analizar los datos correspondientes al GRB 170817A. Para esto se utilizaron 2 horas de datos comprendidas entre las 20:55:09 UTC y las 22:55:09 UTC, tiempo en el que el evento estuvo en el campo de visión de HAWC. Para realizar la búsqueda se utilizó el mismo índice espectral reportado por la colaboración de HAWC, -2.5 para una ley de potencias. Los mapas de significancia de una zona alrededor del trigger no mostraron regiones significativas por lo que aún con la mejora en el algoritmo de separación gamma/hadrón no fue posible detectar el GRB.

Posteriormente y dada la forma tan general con la que se diseñó el algoritmo, se procedió a agregar más horas de datos (entre el 15 y 20 de Agosto 2017) y a realizar un análisis de todo el campo de visión para ese tiempo de datos. En el mapa de significancia se encontró una región con 3.96σ y al buscar asociación con las fuentes de TeVCat se encontró que dicha región tiene correspondencia con el blazar Mrk 421.

Se deja como trabajo a futuro realizar un análisis de todo el cielo abarcado por el observatorio HAWC para ver si es posible observar las fuentes conocidas que actualmente sólo son visibles en altas energías. Además se planea repetir la búsqueda de GRBs que se realizó anteriormente en [60].

Apéndice A

Códigos desarrollados para el análisis

En este capítulo se agregan los códigos que se desarrollaron para el análisis de esta tesis. Una vez que la red neuronal ha sido entrenada y guardados los modelos, se procede a utilizar el código para realizar predicciones sobre los datos. El siguiente código lee el archivo .csv con los eventos, transforma los datos mediante las componentes principales y les aplica la red neuronal para obtener las predicciones. Posteriormente guarda las predicciones en archivos .csv.

```
1
2 #!/usr/bin/python3
3 #autor: AbelR
4 import pandas as pd
5 import numpy as np
6 import matplotlib.pyplot as plt
7 from numpy.linalg import eig
8 import keras
9 from keras.models import model_from_json
10 import argparse
11 import os
12 os.environ['TF_CPP_MIN_LOG_LEVEL'] = '2'
13
14 p=argparse.ArgumentParser(description="Aplica la red neuronal a los archivos csv de HAWC
15 ")
16 p.add_argument("--input", "-i", help="archivo de entrada")
17 p.add_argument("--output", "-o", help="nombre de salida")
18
19 args=p.parse_args()
20
21 lista_cols_sim=np.asarray
22     ([11,12,13,14,15,18,19,20,22,23,25,26,27,36,37,40,41,42,43,44,45,
23      47,48,49,50,51,52,53,54,55,63,64,65,66,67,68,69,70,71])
24 """lista con las columnas a usar de los datos"""
25 cols_datos=np.asarray([11,12,13,14,15,18,19,20,22,23,25,26,27,28,29,32,33,34,
26      35,36,37,39,40,41,42,43,44,45,46,47,55,56,57,58,59,60,
27      61,62,63])
28
29 def estandarizar(df):
30     mean=df.mean()
31     std=df.std()
32     return (df-mean)/std
33
34 def PCA(df_est):
35     """Funcion que realiza un PCA convencional y transforma
36     los datos iniciales a las nuevas variables
37     o componentes principales"""
38     X=np.arange(1,40,1)
39     #Y=np.ones(39)
40     cov_mat=df_est.cov()
41     values,vectors=eig(cov_mat)
42     tot=sum(values)
43     var_exp=[(i/tot)*100 for i in sorted(values,reverse=True)]
44     cum_var_exp=np.cumsum(var_exp)
45     plt.figure(num=1,figsize=(10,10))
46     plt.title("Scree plot")
47     plt.xlabel("Componentes Principales", fontsize=20)
48     plt.ylabel("Varianza explicada", fontsize=20)
49     plt.bar(X,var_exp,color='mediumblue')
50     plt.plot(X,cum_var_exp,'o-',c='limegreen')
51     plt.tick_params(axis='both',labelsize='large')
```

```

50 plt.savefig("/home/abelr/Dropbox/Tareas_Maestria/cosas_tesis/PCA/PCA_completo2.png")
51 #plt.plot(X,Y,linestyle='--',c='r',label='r$\lambda=1$')
52 plt.show()
53
54 idx=0
55 for valor_propio in values:
56     if valor_propio >= 1:
57         idx+=1
58     #print(idx)
59     data_transformed=vectors.T.dot(df_est.T)
60     print("se econcontraron {} valores propios mayores a 1".format(idx))
61     print("-----")
62     print("valor propio",sorted(values,reverse=True))
63     print("Varianza individual",var_exp)
64     print("-----")
65     print("Varianza acumulada", cum_var_exp)
66     return data_transformed.T[:,0:idx], values, vectors
67
68 def load_model(nombre_modelo,nombre_weight,datos):
69     # load json and create model
70     json_file = open('{}'.format(nombre_modelo), 'r')
71     loaded_model_json = json_file.read()
72     json_file.close()
73     loaded_model = model_from_json(loaded_model_json)
74     # load weights into new model
75     loaded_model.load_weights("{}".format(nombre_weight))
76     loaded_model.compile(optimizer='rmsprop',loss='binary_crossentropy',
77                           metrics=['binary_accuracy'])
78     predictions=loaded_model.predict_proba(datos)
79     print("Se cargo el modelo")
80     return predictions
81
82 def predict_data(dir_arch,nombre_salida,cols):
83     """vectors: vectores usados en PCA
84     dir_arch: string con ubicacion del archivo
85     al cual se quiere hacer prediccion
86     nombre_salida: string del archivo de salida
87     cols: columnas que se van a usar del archivo """
88     vectors=np.genfromtxt("/home/abelr/deep_learn_analysis/MAPAS/SWEETS/
89     vectores_pca_finales.txt"
90                           ,unpack=True)
91     df2=pd.read_csv("{}".format(dir_arch),usecols=cols) #explicito si tiene o no headers
92     #df2=np.genfromtxt("{}".format(dir_arch),usecols=cols,delimiter=",")
93     #df2.to_csv("prueba111.csv")
94     df2_est=estandarizar(df2)
95     data_transf=vectors.dot(df2_est.T)
96     # data_transf=vectors.T.dot(df2_est.T) el programa los lee transpuestos
97     data_transf_final=data_transf.T[:,0:9]
98     #print(len(data_transf_final),len(df2),df2.shape)
99     Ysimnew=load_model("/home/abelr/deep_learn_analysis/MAPAS/SWEETS/
100     model_G_H_alldec_4lyr_16un_fin.json",
101                       "/home/abelr/deep_learn_analysis/MAPAS/SWEETS/
102     model_G_H_alldec_4lyr_16un_fin.h5",
103                       data_transf_final)
104     plt.hist(Ysimnew,bins=20,alpha=0.5)
105     plt.savefig("{}_png".format(nombre_salida))
106     plt.clf()
107     mask=Ysimnew>=0.6
108     print(len(Ysimnew[mask]))
109     np.savetxt("{}".format(nombre_salida),Ysimnew)
110
111 print("{}".format(args.input))
112 predict_data("{}".format(args.input),"{}".format(args.output),cols_datos)

```

La forma de utilizar el código es como sigue:

```

python3 red_neuronal.py -i "archivo-csv-de-eventos.csv" -o "archivo-csv-de-predicciones.csv"

```

Una vez que se han creado las predicciones, se procede a crear los nuevos archivos de simulación con el siguiente código:

```

1 //autor: Abel Rosales
2 /*

```

```

3 Este codigo lee un archivo csv con todas las variables de un archivo reco, lee el otro
  csv con la clasificacion
4 de redes neuronales y crea un archivo xcd con todas las entradas. Es necesario modificar
  los iteradores i,j,l dependiendo
5 del numero de eventos que se quieran manejar
6 Funciona solo con archivos de datos. usar el de simulacion para simulacion
7 12/mar/2019 se agregaron lineas para que el programa se salte los headers del archivo
  csv de entrada
8 string header;
9 getline(archivo_datos,header);
10 */
11 #include <xcdf/XCDF.h>
12 #include <hawcnest/CommandLineConfigurator.h>
13 #include <boost/tokenizer.hpp>
14 #include <fstream>
15 #include <cstdio>
16 #include <iostream>
17 #include <vector>
18 #include <string>
19 using namespace std;
20 int main(int argc, char **argv) {
21
22     CommandLineConfigurator cl;
23
24     cl.AddOption<string>("datafile", "data file name");
25     cl.AddOption<string>("predfile", "prediction file name");
26     cl.AddOption<string>("newxcdf", "Name of new XDCF File");
27     cl.AddOption<int64_t>("counts", "Number of events in the file");
28
29     if (!cl.ParseCommandLine(argc, argv))
30         return 1;
31
32     string datafile= cl.GetArgument<string>("datafile");
33     int64_t event_number=cl.GetArgument<int64_t>("counts");
34
35     ifstream archivo_datos(datafile.c_str());
36
37     if(!archivo_datos.is_open()) std::cout << "ERROR: File not Open" << '\n';
38 //vectores usados para guardar los datos del archivo CSV
39     vector<unsigned int> rec_status,rec_version,rec_eventID,rec_runID,rec_timeSliceID,
      rec_trigger_flags,rec_event_flags;
40     vector<unsigned int> rec_gtc_flags,rec_gpsSec,rec_gpsNanosec,rec_nChTot,rec_nChAvail,
      rec_nHitTot,rec_nHit,rec_nHitSP10;
41     vector<unsigned int> rec_nHitSP20,rec_nTankTot,rec_nTankAvail,rec_nTankHitTot,
      rec_nTankHit,rec_windowHits,rec_angleFitStatus;
42     vector<unsigned int> rec_planeNDOF,rec_SFCFNDOF,rec_coreFitStatus,rec_CxPE40PMT,
      rec_CxPE40XnCh,rec_coreFiduScale;
43     vector<float> rec_zenithAngle,rec_azimuthAngle,rec_dec,rec_ra,rec_planeChi2,rec_coreX,
      rec_coreY,rec_logCoreAmplitude;
44     vector<float> rec_coreFitUnc,rec_SFCFChi2,rec_logNNEnergy,rec_fAnnulusCharge0,
      rec_fAnnulusCharge1,rec_fAnnulusCharge2;
45     vector<float> rec_fAnnulusCharge3,rec_fAnnulusCharge4,rec_fAnnulusCharge5,
      rec_fAnnulusCharge6,rec_fAnnulusCharge7;
46     vector<float> rec_fAnnulusCharge8,rec_protonlheEnergy,rec_protonlheLLH,
      rec_gammalheEnergy,rec_gammalheLLH;
47     vector<float> rec_chargeFiduScale50,rec_chargeFiduScale70,rec_chargeFiduScale90,
      rec_logMaxPE,rec_logNPE;
48     vector<float> rec_CxPE40,rec_CxPE40SPTIME,rec_LDFAge,rec_LDFamp,rec_LDFChi2,rec_PINC,
      rec_disMax;
49     vector<float> class_labels;
50
51 //nombres temporales donde guardar los datos
52
53     string temprec_status,temprec_version,temprec_eventID,temprec_runID,
      temprec_timeSliceID,temprec_trigger_flags,temprec_event_flags;
54     string temprec_gtc_flags,temprec_gpsSec,temprec_gpsNanosec,temprec_nChTot,
      temprec_nChAvail,temprec_nHitTot,temprec_nHit,temprec_nHitSP10;
55     string temprec_nHitSP20,temprec_nTankTot,temprec_nTankAvail,temprec_nTankHitTot,
      temprec_nTankHit,temprec_windowHits,temprec_angleFitStatus;
56     string temprec_planeNDOF,temprec_SFCFNDOF,temprec_coreFitStatus,temprec_CxPE40PMT,
      temprec_CxPE40XnCh,temprec_coreFiduScale;
57
58     string temprec_zenithAngle,temprec_azimuthAngle,temprec_dec,temprec_ra,
      temprec_planeChi2,temprec_coreX;
59     string temprec_coreY,temprec_logCoreAmplitude,temprec_coreFitUnc,temprec_SFCFChi2,
      temprec_logNNEnergy;

```

```

60 string temprec_fAnnulusCharge0,temprec_fAnnulusCharge1,temprec_fAnnulusCharge2,
    temprec_fAnnulusCharge3;
61 string temprec_fAnnulusCharge4,temprec_fAnnulusCharge5,temprec_fAnnulusCharge6,
    temprec_fAnnulusCharge7;
62 string temprec_fAnnulusCharge8,temprec_protonlheEnergy,temprec_protonlheLLH,
    temprec_gammalheEnergy,temprec_gammalheLLH;
63 string temprec_chargeFiduScale50,temprec_chargeFiduScale70,temprec_chargeFiduScale90,
    temprec_logMaxPE,temprec_logNPE;
64 string temprec_CxPE40,temprec_CxPE40SPTIME,temprec_LDFAge,temprec_LDFamp,
    temprec_LDFChi2,temprec_PINC,temprec_disMax;
65 string tempclass_labels;
66 int i=0;
67 int j=0;
68 string header;
69 getline(archivo_datos,header);
70 while(archivo_datos.good()){
71
72     // se lee la línea hasta el separador "," y se guarda en la variable temporal
73     getline(archivo_datos,temprec_status,',');
74     getline(archivo_datos,temprec_version,',');
75     getline(archivo_datos,temprec_eventID,',');
76     getline(archivo_datos,temprec_runID,',');
77     getline(archivo_datos,temprec_timeSliceID,',');
78     getline(archivo_datos,temprec_trigger_flags,',');
79     getline(archivo_datos,temprec_event_flags,',');
80     getline(archivo_datos,temprec_gtc_flags,',');
81     getline(archivo_datos,temprec_gpsSec,',');
82     getline(archivo_datos,temprec_gpsNanosec,',');
83     getline(archivo_datos,temprec_nChTot,',');
84     getline(archivo_datos,temprec_nChAvail,',');
85     getline(archivo_datos,temprec_nHitTot,',');
86     getline(archivo_datos,temprec_nHit,',');
87     getline(archivo_datos,temprec_nHitSP10,',');
88     getline(archivo_datos,temprec_nHitSP20,',');
89     getline(archivo_datos,temprec_nTankTot,',');
90     getline(archivo_datos,temprec_nTankAvail,',');
91     getline(archivo_datos,temprec_nTankHitTot,',');
92     getline(archivo_datos,temprec_nTankHit,',');
93     getline(archivo_datos,temprec_windowHits,',');
94     getline(archivo_datos,temprec_angleFitStatus,',');
95     getline(archivo_datos,temprec_planeNDOF,',');
96     getline(archivo_datos,temprec_SFCFNDOF,',');
97     getline(archivo_datos,temprec_coreFitStatus,',');
98     getline(archivo_datos,temprec_CxPE40PMT,',');
99     getline(archivo_datos,temprec_CxPE40XnCh,',');
100    getline(archivo_datos,temprec_coreFiduScale,',');
101    getline(archivo_datos,temprec_zenithAngle,',');
102    getline(archivo_datos,temprec_azimuthAngle,',');
103    getline(archivo_datos,temprec_dec,',');
104    getline(archivo_datos,temprec_ra,',');
105    getline(archivo_datos,temprec_planeChi2,',');
106    getline(archivo_datos,temprec_coreX,',');
107    getline(archivo_datos,temprec_coreY,',');
108    getline(archivo_datos,temprec_logCoreAmplitude,',');
109    getline(archivo_datos,temprec_coreFitUnc,',');
110    getline(archivo_datos,temprec_SFCFChi2,',');
111    getline(archivo_datos,temprec_logNNEnergy,',');
112    getline(archivo_datos,temprec_fAnnulusCharge0,',');
113    getline(archivo_datos,temprec_fAnnulusCharge1,',');
114    getline(archivo_datos,temprec_fAnnulusCharge2,',');
115    getline(archivo_datos,temprec_fAnnulusCharge3,',');
116    getline(archivo_datos,temprec_fAnnulusCharge4,',');
117    getline(archivo_datos,temprec_fAnnulusCharge5,',');
118    getline(archivo_datos,temprec_fAnnulusCharge6,',');
119    getline(archivo_datos,temprec_fAnnulusCharge7,',');
120    getline(archivo_datos,temprec_fAnnulusCharge8,',');
121    getline(archivo_datos,temprec_protonlheEnergy,',');
122    getline(archivo_datos,temprec_protonlheLLH,',');
123    getline(archivo_datos,temprec_gammalheEnergy,',');
124    getline(archivo_datos,temprec_gammalheLLH,',');
125    getline(archivo_datos,temprec_chargeFiduScale50,',');
126    getline(archivo_datos,temprec_chargeFiduScale70,',');
127    getline(archivo_datos,temprec_chargeFiduScale90,',');
128    getline(archivo_datos,temprec_logMaxPE,',');
129    getline(archivo_datos,temprec_logNPE,',');
130    getline(archivo_datos,temprec_CxPE40,',');

```

```

131     getline(archivo_datos , temprec_CxPE40SPTime , ',' , ');
132     getline(archivo_datos , temprec_LDFAge , ',' , ');
133     getline(archivo_datos , temprec_LDFamp , ',' , ');
134     getline(archivo_datos , temprec_LDFChi2 , ',' , ');
135     getline(archivo_datos , temprec_PINC , ',' , ');
136     getline(archivo_datos , temprec_disMax , '\n' );
137     i=i+1;
138     if(i>event_number) break;
139 //convertimos el string a unsigned int y a float segun el caso
140 // y almacenamos en los vectores
141
142     rec_status.push_back(stoul(temprec_status));
143     rec_version.push_back(stoul(temprec_version));
144     rec_eventID.push_back(stoul(temprec_eventID));
145     rec_runID.push_back(stoul(temprec_runID));
146     rec_timeSliceID.push_back(stoul(temprec_timeSliceID));
147     rec_trigger_flags.push_back(stoul(temprec_trigger_flags));
148     rec_event_flags.push_back(stoul(temprec_event_flags));
149     rec_gtc_flags.push_back(stoul(temprec_gtc_flags));
150     rec_gpsSec.push_back(stoul(temprec_gpsSec));
151     rec_gpsNanosec.push_back(stoul(temprec_gpsNanosec));
152     rec_nChTot.push_back(stoul(temprec_nChTot));
153     rec_nChAvail.push_back(stoul(temprec_nChAvail));
154     rec_nHitTot.push_back(stoul(temprec_nHitTot));
155     rec_nHit.push_back(stoul(temprec_nHit));
156     rec_nHitSP10.push_back(stoul(temprec_nHitSP10)); //enteros
157     rec_nHitSP20.push_back(stoul(temprec_nHitSP20));
158     rec_nTankTot.push_back(stoul(temprec_nTankTot));
159     rec_nTankAvail.push_back(stoul(temprec_nTankAvail));
160     rec_nTankHitTot.push_back(stoul(temprec_nTankHitTot));
161     rec_nTankHit.push_back(stoul(temprec_nTankHit));
162     rec_windowHits.push_back(stoul(temprec_windowHits));
163     rec_angleFitStatus.push_back(stoul(temprec_angleFitStatus));
164     rec_planeNDOF.push_back(stoul(temprec_planeNDOF));
165     rec_SF CFNDOF.push_back(stoul(temprec_SF CFNDOF));
166     rec_coreFitStatus.push_back(stoul(temprec_coreFitStatus));
167     rec_CxPE40PMT.push_back(stoul(temprec_CxPE40PMT));
168     rec_CxPE40XnCh.push_back(stoul(temprec_CxPE40XnCh));
169     rec_coreFiduScale.push_back(stoul(temprec_coreFiduScale));
170     rec_zenithAngle.push_back(stof(temprec_zenithAngle)); //floats
171     rec_azimuthAngle.push_back(stof(temprec_azimuthAngle));
172     rec_dec.push_back(stof(temprec_dec));
173     rec_ra.push_back(stof(temprec_ra));
174     rec_planeChi2.push_back(stof(temprec_planeChi2));
175     rec_coreX.push_back(stof(temprec_coreX));
176     rec_coreY.push_back(stof(temprec_coreY));
177     rec_logCoreAmplitude.push_back(stof(temprec_logCoreAmplitude));
178     rec_coreFitUnc.push_back(stof(temprec_coreFitUnc));
179     rec_SF CFChi2.push_back(stof(temprec_SF CFChi2));
180     rec_logNNEnergy.push_back(stof(temprec_logNNEnergy));
181     rec_fAnnulusCharge0.push_back(stof(temprec_fAnnulusCharge0));
182     rec_fAnnulusCharge1.push_back(stof(temprec_fAnnulusCharge1));
183     rec_fAnnulusCharge2.push_back(stof(temprec_fAnnulusCharge2));
184     rec_fAnnulusCharge3.push_back(stof(temprec_fAnnulusCharge3));
185     rec_fAnnulusCharge4.push_back(stof(temprec_fAnnulusCharge4));
186     rec_fAnnulusCharge5.push_back(stof(temprec_fAnnulusCharge5));
187     rec_fAnnulusCharge6.push_back(stof(temprec_fAnnulusCharge6));
188     rec_fAnnulusCharge7.push_back(stof(temprec_fAnnulusCharge7));
189     rec_fAnnulusCharge8.push_back(stof(temprec_fAnnulusCharge8));
190     rec_protonlheEnergy.push_back(stof(temprec_protonlheEnergy));
191     rec_protonlheLLH.push_back(stof(temprec_protonlheLLH));
192     rec_gammalheEnergy.push_back(stof(temprec_gammalheEnergy));
193     rec_gammalheLLH.push_back(stof(temprec_gammalheLLH));
194     rec_chargeFiduScale50.push_back(stof(temprec_chargeFiduScale50));
195     rec_chargeFiduScale70.push_back(stof(temprec_chargeFiduScale70));
196     rec_chargeFiduScale90.push_back(stof(temprec_chargeFiduScale90));
197     rec_logMaxPE.push_back(stof(temprec_logMaxPE));
198     rec_logNPE.push_back(stof(temprec_logNPE));
199     rec_CxPE40.push_back(stof(temprec_CxPE40));
200     rec_CxPE40SPTime.push_back(stof(temprec_CxPE40SPTime));
201     rec_LDFAge.push_back(stof(temprec_LDFAge));
202     rec_LDFamp.push_back(stof(temprec_LDFamp));
203     rec_LDFChi2.push_back(stof(temprec_LDFChi2));
204     rec_PINC.push_back(stof(temprec_PINC));
205     rec_disMax.push_back(stof(temprec_disMax));
206

```

```

207     }
208     cout << rec_disMax.size() << endl;
209     archivo_datos.close();
210     //leemos labels
211     string predfile= cl.GetArgument<string>("predfile");
212     ifstream archivo_labels(predfile.c_str());
213     if(!archivo_labels.is_open()) std::cout << "ERROR: File Open" << '\n';
214     while(archivo_labels.good()){
215         getline(archivo_labels, tempclass_labels, '\n');
216         j=j+1;
217         if(j>event_number) break;
218         class_labels.push_back(stof(tempclass_labels));
219         //cout << tempclass_labels << endl;
220
221     }
222     cout << class_labels.size()<<endl;
223     archivo_labels.close();
224     //for(int k=0;k<10;k++){
225     //cout<<rec_LDFChi2[k]<<" "<<rec_disMax[k]<<" "<<rec_PINC[k]<<endl;
226
227     //}
228
229     std::cout << "Creating xcdf file" << std::endl;
230     string newxcdf= cl.GetArgument<string>("newxcdf");
231     XCDFFile f1(newxcdf.c_str(),"w");
232     //creamos los campos
233     XCDFUnsignedIntegerField raw_status =
234         f1.AllocateUnsignedIntegerField("rec.status", 1);
235     XCDFUnsignedIntegerField raw_version =
236         f1.AllocateUnsignedIntegerField("rec.version", 1);
237     XCDFUnsignedIntegerField raw_eventID =
238         f1.AllocateUnsignedIntegerField("rec.eventID", 1);
239     XCDFUnsignedIntegerField raw_runID =
240         f1.AllocateUnsignedIntegerField("rec.runID", 1);
241     XCDFUnsignedIntegerField raw_timeSliceID =
242         f1.AllocateUnsignedIntegerField("rec.timeSliceID", 1);
243     XCDFUnsignedIntegerField raw_trigger_flags =
244         f1.AllocateUnsignedIntegerField("rec.trigger_flags", 1);
245     XCDFUnsignedIntegerField raw_event_flags =
246         f1.AllocateUnsignedIntegerField("rec.event_flags", 1);
247     XCDFUnsignedIntegerField raw_gtc_flags =
248         f1.AllocateUnsignedIntegerField("rec.gtc_flags", 1);
249     XCDFUnsignedIntegerField raw_gpsSec =
250         f1.AllocateUnsignedIntegerField("rec.gpsSec", 1);
251     XCDFUnsignedIntegerField raw_gpsNanosec =
252         f1.AllocateUnsignedIntegerField("rec.gpsNanosec", 1);
253     XCDFUnsignedIntegerField raw_nChTot =
254         f1.AllocateUnsignedIntegerField("rec.nChTot", 1);
255     XCDFUnsignedIntegerField raw_nChAvail =
256         f1.AllocateUnsignedIntegerField("rec.nChAvail", 1);
257     XCDFUnsignedIntegerField raw_nHitTot =
258         f1.AllocateUnsignedIntegerField("rec.nHitTot", 1);
259     XCDFUnsignedIntegerField raw_nHit =
260         f1.AllocateUnsignedIntegerField("rec.nHit", 1);
261     XCDFUnsignedIntegerField raw_nHitSP10 =
262         f1.AllocateUnsignedIntegerField("rec.nHitSP10", 1);
263     XCDFUnsignedIntegerField raw_nHitSP20 =
264         f1.AllocateUnsignedIntegerField("rec.nHitSP20", 1);
265     XCDFUnsignedIntegerField raw_nTankTot =
266         f1.AllocateUnsignedIntegerField("rec.nTankTot", 1);
267     XCDFUnsignedIntegerField raw_nTankAvail =
268         f1.AllocateUnsignedIntegerField("rec.nTankAvail", 1);
269     XCDFUnsignedIntegerField raw_nTankHitTot =
270         f1.AllocateUnsignedIntegerField("rec.nTankHitTot", 1);
271     XCDFUnsignedIntegerField raw_nTankHit =
272         f1.AllocateUnsignedIntegerField("rec.nTankHit", 1);
273     XCDFUnsignedIntegerField raw_windowHits =
274         f1.AllocateUnsignedIntegerField("rec.windowHits", 1);
275     XCDFUnsignedIntegerField raw_angleFitStatus =
276         f1.AllocateUnsignedIntegerField("rec.angleFitStatus", 1);
277     XCDFUnsignedIntegerField raw_planeNDOF =
278         f1.AllocateUnsignedIntegerField("rec.planeNDOF", 1);
279     XCDFUnsignedIntegerField raw_SFCFNDOF =
280         f1.AllocateUnsignedIntegerField("rec.SFCFNDOF", 1);
281     XCDFUnsignedIntegerField raw_coreFitStatus =
282         f1.AllocateUnsignedIntegerField("rec.coreFitStatus", 1);

```

```
283 XCDFUnsignedIntegerField raw_CxPE40PMT =
284     f1.AllocateUnsignedIntegerField("rec.CxPE40PMT", 1);
285 XCDFUnsignedIntegerField raw_CxPE40XnCh =
286     f1.AllocateUnsignedIntegerField("rec.CxPE40XnCh", 1);
287 XCDFUnsignedIntegerField raw_coreFiduScale =
288     f1.AllocateUnsignedIntegerField("rec.coreFiduScale", 1);
289 XCDFFloatingPointField raw_zenithAngle =
290     f1.AllocateFloatingPointField("rec.zenithAngle", 0.0001);
291 XCDFFloatingPointField raw_azimuthAngle =
292     f1.AllocateFloatingPointField("rec.azimuthAngle", 0.0001);
293 XCDFFloatingPointField raw_dec =
294     f1.AllocateFloatingPointField("rec.dec", 0.0001);
295 XCDFFloatingPointField raw_ra =
296     f1.AllocateFloatingPointField("rec.ra", 0.0001);
297 XCDFFloatingPointField raw_planeChi2 =
298     f1.AllocateFloatingPointField("rec.planeChi2", 0.01);
299 XCDFFloatingPointField raw_coreX =
300     f1.AllocateFloatingPointField("rec.coreX", 0.1);
301 XCDFFloatingPointField raw_coreY =
302     f1.AllocateFloatingPointField("rec.coreY", 0.1);
303 XCDFFloatingPointField raw_logCoreAmplitude =
304     f1.AllocateFloatingPointField("rec.logCoreAmplitude", 0.1);
305 XCDFFloatingPointField raw_coreFitUnc =
306     f1.AllocateFloatingPointField("rec.coreFitUnc", 0.1);
307 XCDFFloatingPointField raw_SFChi2 =
308     f1.AllocateFloatingPointField("rec.SFChi2", 0.01);
309 XCDFFloatingPointField raw_logNNEnergy =
310     f1.AllocateFloatingPointField("rec.logNNEnergy", 0.01);
311 XCDFFloatingPointField raw_fAnnulusCharge0 =
312     f1.AllocateFloatingPointField("rec.fAnnulusCharge0", 0.1);
313 XCDFFloatingPointField raw_fAnnulusCharge1 =
314     f1.AllocateFloatingPointField("rec.fAnnulusCharge1", 0.01);
315 XCDFFloatingPointField raw_fAnnulusCharge2 =
316     f1.AllocateFloatingPointField("rec.fAnnulusCharge2", 0.01);
317 XCDFFloatingPointField raw_fAnnulusCharge3 =
318     f1.AllocateFloatingPointField("rec.fAnnulusCharge3", 0.01);
319 XCDFFloatingPointField raw_fAnnulusCharge4 =
320     f1.AllocateFloatingPointField("rec.fAnnulusCharge4", 0.01);
321 XCDFFloatingPointField raw_fAnnulusCharge5 =
322     f1.AllocateFloatingPointField("rec.fAnnulusCharge5", 0.01);
323 XCDFFloatingPointField raw_fAnnulusCharge6 =
324     f1.AllocateFloatingPointField("rec.fAnnulusCharge6", 0.01);
325 XCDFFloatingPointField raw_fAnnulusCharge7 =
326     f1.AllocateFloatingPointField("rec.fAnnulusCharge7", 0.01);
327 XCDFFloatingPointField raw_fAnnulusCharge8 =
328     f1.AllocateFloatingPointField("rec.fAnnulusCharge8", 0.01);
329 XCDFFloatingPointField raw_protonlheEnergy =
330     f1.AllocateFloatingPointField("rec.protonlheEnergy", 0.01);
331 XCDFFloatingPointField raw_protonlheLLH =
332     f1.AllocateFloatingPointField("rec.protonlheLLH", 0.01);
333 XCDFFloatingPointField raw_gammalheEnergy =
334     f1.AllocateFloatingPointField("rec.gammalheEnergy", 0.01);
335 XCDFFloatingPointField raw_gammalheLLH =
336     f1.AllocateFloatingPointField("rec.gammalheLLH", 0.01);
337 XCDFFloatingPointField raw_chargeFiduScale50 =
338     f1.AllocateFloatingPointField("rec.chargeFiduScale50", 0.01);
339 XCDFFloatingPointField raw_chargeFiduScale70 =
340     f1.AllocateFloatingPointField("rec.chargeFiduScale70", 0.01);
341 XCDFFloatingPointField raw_chargeFiduScale90 =
342     f1.AllocateFloatingPointField("rec.chargeFiduScale90", 0.01);
343 XCDFFloatingPointField raw_logMaxPE =
344     f1.AllocateFloatingPointField("rec.logMaxPE", 0.01);
345 XCDFFloatingPointField raw_logNPE =
346     f1.AllocateFloatingPointField("rec.logNPE", 0.01);
347 XCDFFloatingPointField raw_CxPE40 =
348     f1.AllocateFloatingPointField("rec.CxPE40", 0.01);
349 XCDFFloatingPointField raw_CxPE40SPTime =
350     f1.AllocateFloatingPointField("rec.CxPE40SPTime", 0.1);
351 XCDFFloatingPointField raw_LDFAge =
352     f1.AllocateFloatingPointField("rec.LDFAge", 0.01);
353 XCDFFloatingPointField raw_LDFamp =
354     f1.AllocateFloatingPointField("rec.LDFamp", 0.01);
355 XCDFFloatingPointField raw_LDFChi2 =
356     f1.AllocateFloatingPointField("rec.LDFChi2", 0.01);
357 XCDFFloatingPointField raw_PINC =
358     f1.AllocateFloatingPointField("rec.PINC", 0.01);
```



```

359   XCDFFloatingPointField raw_disMax =
360       f1.AllocateFloatingPointField("rec.disMax", 0.01);
361   XCDFFloatingPointField raw_NNPred =
362       f1.AllocateFloatingPointField("rec.NNPred", 0.001);
363   //asignamos eventos al xcdf
364   for(int l = 0;l<event_number;l++){
365       raw_status << rec_status[l];
366
367       raw_version << rec_version[l];
368
369       raw_eventID << rec_eventID[l];
370
371       raw_runID <<rec_runID[l];
372
373       raw_timeSliceID << rec_timeSliceID[l];
374
375       raw_trigger_flags << rec_trigger_flags[l];
376
377       raw_event_flags << rec_event_flags[l];
378
379       raw_gtc_flags << rec_gtc_flags[l];
380
381       raw_gpsSec << rec_gpsSec[l];
382
383       raw_gpsNanosec << rec_gpsNanosec[l];
384
385       raw_nChTot << rec_nChTot[l];
386
387       raw_nChAvail << rec_nChAvail[l];
388
389       raw_nHitTot << rec_nHitTot[l];
390
391       raw_nHit << rec_nHit[l];
392
393       raw_nHitSP10 << rec_nHitSP10[l];
394
395       raw_nHitSP20 << rec_nHitSP20[l];
396
397       raw_nTankTot << rec_nTankTot[l];
398
399       raw_nTankAvail << rec_nTankAvail[l];
400
401       raw_nTankHitTot << rec_nTankHitTot[l];
402
403       raw_nTankHit << rec_nTankHit[l];
404
405       raw_windowHits << rec_windowHits[l];
406
407       raw_angleFitStatus << rec_angleFitStatus[l];
408
409       raw_planeNDOF << rec_planeNDOF[l];
410
411       raw_SFCFNDOF << rec_SFCFNDOF[l];
412
413       raw_coreFitStatus << rec_coreFitStatus[l];
414
415       raw_CxPE40PMT << rec_CxPE40PMT[l];
416
417       raw_CxPE40XnCh << rec_CxPE40XnCh[l];
418
419       raw_coreFiduScale << rec_coreFiduScale[l];
420
421       raw_zenithAngle << rec_zenithAngle[l];
422
423       raw_azimuthAngle << rec_azimuthAngle[l];
424
425       raw_dec << rec_dec[l];
426
427       raw_ra << rec_ra[l];
428
429       raw_planeChi2 << rec_planeChi2[l];
430
431       raw_coreX << rec_coreX[l];
432
433       raw_coreY << rec_coreY[l];
434

```

```

435     raw_logCoreAmplitude << rec_logCoreAmplitude[1];
436
437     raw_coreFitUnc << rec_coreFitUnc[1];
438
439     raw_SFChi2 << rec_SFChi2[1];
440
441     raw_logNNEnergy << rec_logNNEnergy[1];
442
443     raw_fAnnulusCharge0 << rec_fAnnulusCharge0[1];
444
445     raw_fAnnulusCharge1 << rec_fAnnulusCharge1[1];
446
447     raw_fAnnulusCharge2 << rec_fAnnulusCharge2[1];
448
449     raw_fAnnulusCharge3 << rec_fAnnulusCharge3[1];
450
451     raw_fAnnulusCharge4 << rec_fAnnulusCharge4[1];
452
453     raw_fAnnulusCharge5 << rec_fAnnulusCharge5[1];
454
455     raw_fAnnulusCharge6 << rec_fAnnulusCharge6[1];
456
457     raw_fAnnulusCharge7 << rec_fAnnulusCharge7[1];
458
459     raw_fAnnulusCharge8 << rec_fAnnulusCharge8[1];
460
461     raw_protonlheEnergy << rec_protonlheEnergy[1];
462
463     raw_protonlheLLH << rec_protonlheLLH[1];
464
465     raw_gammalheEnergy << rec_gammalheEnergy[1];
466
467     raw_gammalheLLH << rec_gammalheLLH[1];
468
469     raw_chargeFiduScale50 << rec_chargeFiduScale50[1];
470
471     raw_chargeFiduScale70 << rec_chargeFiduScale70[1];
472
473     raw_chargeFiduScale90 << rec_chargeFiduScale90[1];
474
475     raw_logMaxPE << rec_logMaxPE[1];
476
477     raw_logNPE << rec_logNPE[1];
478
479     raw_CxPE40 << rec_CxPE40[1];
480
481     raw_CxPE40SPTime << rec_CxPE40SPTime[1];
482
483     raw_LDFAge << rec_LDFAge[1];
484
485     raw_LDFamp << rec_LDFamp[1];
486
487     raw_LDFChi2 << rec_LDFChi2[1];
488
489     raw_PINC << rec_PINC[1];
490
491     raw_disMax << rec_disMax[1];
492
493     raw_NNPred << class_labels[1];
494
495
496
497     f1.Write();
498 }
499 f1.Close();
500
501
502 }
```

El código entonces se compila de manera usual. A continuación la forma de ejecutarlo desde la terminal:

```

xcdf_data_header --datafile "archivo-csv-de-eventos.csv" --predfile "archivo-de-
predicciones.csv" --newxcdf "nombre-del-xcdf-de-salida.xcd" --counts "numero-de-
eventos-en-el-archivo-de-eventos"
```

Dado que se trabajó con miles de archivos, los comandos anteriores de bash tuvieron que ser automati-

zados. A continuación un ejemplo sobre un tránsito.

```
for i in {069..221..1}; do python3 ../red_neuronal.py -i anglefit0_2_3_run007183-00$i.csv -o predictions_subruns$i.csv; done
```

Y para crear los archivos xcd:

```
for i in {069..221..1}; do
input_csv="anglefit0_2_3_run007183-00$i.csv"
input_pred="predictions_subruns$i.csv"
output_xcd="reco-run007183_00$i.xcd"
cuentas_ar=$(wc -l < "anglefit0_2_3_run007183-00$i.csv") #contamos el numero de eventos
en cada archivo csv
echo $((cuentas_ar-1))
/home/abelr/deep_learn_analysis/MAPAS/xcdf_data_header --datafile $input_csv --predfile
$input_pred --newxcdf $output_xcd --counts $((cuentas_ar-1))
echo "$input_csv","$input_pred","$output_xcd" #para verificar que se estan usando
archivos compatibles
printf "\n"
done
```

Referencias

- [1] Enrico Costa and Filippo Frontera. Gamma Ray Burst origin and their afterglow: story of a discovery and more. *arXiv e-prints*, Jul 2011.
- [2] M Ackermann, M Ajello, K Asano, WB Atwood, Magnus Axelsson, Luca Baldini, J Ballet, G Barbiellini, MG Baring, Dea Bastieri, et al. Fermi-lat observations of the gamma-ray burst grb 130427a. *Science*, 343(6166):42–47, 2014.
- [3] Edo Berger. Short-duration gamma-ray bursts. *Annual Review of Astronomy and Astrophysics*, 52:43–105, 2014.
- [4] Ray W Klebesadel, Ian B Strong, and Roy A Olson. Observations of gamma-ray bursts of cosmic origin. *The Astrophysical Journal*, 182:L85, 1973.
- [5] Compton gamma ray observatory. <https://heasarc.gsfc.nasa.gov/docs/cgro/cgro/>, 1991-2000. [Online; accedida por última vez en Marzo 2019].
- [6] Current gamma-ray burst catalog. <https://gammaray.msfc.nasa.gov/batse/grb/catalog/current/index.html>, 2000. [Online; accedida por última vez en Febrero 2019].
- [7] Charles A Meegan, Geoffrey N Pendleton, Michael S Briggs, Chryssa Kouveliotou, Thomas M Koshut, John Patrick Lestrade, William S Paciesas, Michael L McCollough, Jerome J Brainerd, John M Horack, et al. The third BATSE gamma-ray burst catalog. *The Astrophysical Journal Supplement Series*, 106:65–110, 1996.
- [8] Katharine C. Walker, Bradley E. Schaefer, and E. E. Fenimore. Gamma-ray bursts have millisecond variability. *The Astrophysical Journal*, 537(1):264–269, jul 2000.
- [9] D Band, J Matteson, L Ford, B Schaefer, D Palmer, B Teegarden, T Cline, M Briggs, W Paciesas, G Pendleton, et al. BATSE observations of gamma-ray burst spectra. i-spectral diversity. *The Astrophysical Journal*, 413:281–292, 1993.
- [10] MS Briggs, DL Band, RM Kippen, RD Preece, C Kouveliotou, J Van Paradijs, GH Share, RJ Murphy, SM Matz, A Connors, et al. Observations of GRB 990123 by the compton gamma ray observatory. *The Astrophysical Journal*, 524(1):82, 1999.
- [11] William S Paciesas, Charles A Meegan, Geoffrey N Pendleton, Michael S Briggs, Chryssa Kouveliotou, Thomas M Koshut, John Patrick Lestrade, Michael L McCollough, Jerome J Brainerd, Jon Hakkila, et al. The fourth BATSE gamma-ray burst catalog (revised). *The Astrophysical Journal Supplement Series*, 122(2):465, 1999.
- [12] E. Costa, F. Frontera, J. Heise, M. Feroci, J. in't Zand, F. Fiore, M. N. Cinti, D. Dal Fiume, L. Nicastro, M. Orlandini, E. Palazzi, G. Zavattini, R. Jager, A. Parmar, A. Owens, S. Molendi, G. Cusumano, M. C. Maccarone, S. Giarrusso, A. Coletta, L. A. Antonelli, P. Giommi, J. M. Muller, L. Piro, and R. C. Butler. Discovery of an X-ray afterglow associated with the γ -ray burst of 28 February 1997. *Nature*, 387:783–785, Jun 1997.
- [13] J Van Paradijs, PJ Groot, T Galama, C Kouveliotou, RG Strom, J Telting, RGM Rutten, GJ Fishman, CA Meegan, M Pettini, et al. Transient optical emission from the error box of the γ -ray burst of 28 february 1997. *Nature*, 386(6626):686, 1997.
- [14] M. R. Metzger, S. G. Djorgovski, S. R. Kulkarni, C. C. Steidel, K. L. Adelberger, D. A. Frail, E. Costa, and F. Frontera. Spectral constraints on the redshift of the optical counterpart to the γ -ray burst of 8 May 1997. *Nature*, 387:878–880, June 1997.

- [15] Titus J Galama, PM Vreeswijk, J Van Paradijs, C Kouveliotou, T Augusteijn, H Bönhardt, JP Brewer, V Doublier, J-F Gonzalez, B Leibundgut, et al. An unusual supernova in the error box of the γ -ray burst of 25 april 1998. *Nature*, 395(6703):670, 1998.
- [16] L. Amati, F. Frontera, M. Tavani, J. J. M. in't Zand, A. Antonelli, E. Costa, M. Feroci, C. Guidorzi, J. Heise, N. Masetti, E. Montanari, L. Nicastro, E. Palazzi, E. Pian, L. Piro, and P. Soffitta. Intrinsic spectra and energetics of BeppoSAX Gamma-Ray Bursts with known redshifts. *Astronomy and Astrophysics*, 390:81–89, July 2002.
- [17] G. Cavallo and M. J. Rees. A qualitative study of cosmic fireballs and gamma-ray bursts. *Monthly Notices of the Royal Astronomical Society*, 183(3):359–365, 07 1978.
- [18] PW Guilbert, AC Fabian, and MJ Rees. Spectral and variability constraints on compact sources. *Monthly Notices of the Royal Astronomical Society*, 205(3):593–603, 1983.
- [19] Matthew G Baring and Alice K Harding. The escape of high-energy photons from gamma-ray bursts. *The Astrophysical Journal*, 491(2):663, 1997.
- [20] P Mészáros, Martin J Rees, and H Papatthanassiou. Spectral properties of blast wave models of gamma-ray burst sources. *arXiv preprint astro-ph/9311071*, 1993.
- [21] Bohdan Paczynski and Guohong Xu. Neutrino bursts from gamma-ray bursts. *The Astrophysical Journal*, 427:708–713, 1994.
- [22] MJ Rees and P Mészáros. Relativistic fireballs: energy conversion and time-scales. *Monthly Notices of the Royal Astronomical Society*, 258(1):41P–43P, 1992.
- [23] Ralph AMJ Wijers, Martin J Rees, and Peter Meszaros. Shocked by grb 970228: the afterglow of a cosmological fireball. *Monthly Notices of the Royal Astronomical Society*, 288(4):L51–L56, 1997.
- [24] Tsvi Piran. Gamma-ray bursts and the fireball model. *Physics Reports*, 314(6):575–667, 1999.
- [25] Gamma-ray burst observations. <https://fermi.gsfc.nasa.gov/ssc/observations/types/grbs/>, 2014. [Online; accedida por última vez en Febrero 2019].
- [26] Russell A Hulse and Joseph H Taylor. Discovery of a pulsar in a binary system. *The Astrophysical Journal*, 195:L51–L53, 1975.
- [27] Chris L Fryer, SE Woosley, and Dieter H Hartmann. Formation rates of black hole accretion disk gamma-ray bursts. *The Astrophysical Journal*, 526(1):152, 1999.
- [28] GM Harry. Gm harry (ligo scientific collaboration), classical quantum gravity 27, 084006 (2010). *Classical Quantum Gravity*, 27:084006, 2010.
- [29] James M Lattimer and David N Schramm. Black-hole-neutron-star collisions. *The Astrophysical Journal*, 192:L145–L147, 1974.
- [30] David Eichler, Mario Livio, Tsvi Piran, and David N Schramm. Nucleosynthesis, neutrino bursts and γ -rays from coalescing neutron stars. *Nature*, 340(6229):126, 1989.
- [31] Stephan Rosswog, T Piran, and E Nakar. The multimessenger picture of compact object encounters: binary mergers versus dynamical collisions. *Monthly Notices of the Royal Astronomical Society*, 430(4):2585–2604, 2013.
- [32] M Ruffert and H-Th Janka. Coalescing neutron stars-a step towards physical models-iii. improved numerics and different neutron star masses and spins. *Astronomy & Astrophysics*, 380(2):544–577, 2001.
- [33] Tsvi Piran, Ehud Nakar, and Stephan Rosswog. The electromagnetic signals of compact binary mergers. *Monthly Notices of the Royal Astronomical Society*, 430(3):2121–2136, 2013.
- [34] P D'Avanzo. Short gamma-ray bursts: A review. *Journal of High Energy Astrophysics*, 7:73–80, 2015.
- [35] Scott D Barthelmy, G Chincarini, DN Burrows, N Gehrels, S Covino, A Moretti, P Romano, PT O'Brien, CL Sarazin, C Kouveliotou, et al. An origin for short γ -ray bursts unassociated with current star formation. *Nature*, 438(7070):994, 2005.
- [36] Edo Berger. The prompt gamma-ray and afterglow energies of short-duration gamma-ray bursts. *The Astrophysical Journal*, 670(2):1254, 2007.

- [37] Rudy C Gilmore, Piero Madau, Joel R Primack, Rachel S Somerville, and Francesco Haardt. GeV gamma-ray attenuation and the high-redshift UV background. *Monthly Notices of the Royal Astronomical Society*, 399(4):1694–1708, 2009.
- [38] Neil Gehrels, Carl E Fichtel, Gerald J Fishman, James D Kurfess, and Volker Schönfelder. The Compton gamma ray observatory. *Scientific American*, 269(6):68–77, 1993.
- [39] WB Atwood, Aous A Abdo, Markus Ackermann, W Althouse, B Anderson, M Axelsson, Luca Baldini, J Ballet, DL Band, Guido Barbiellini, et al. The Large Area Telescope on the Fermi gamma-ray space telescope mission. *The Astrophysical Journal*, 697(2):1071, 2009.
- [40] Jim Hinton. Ground-based gamma-ray astronomy with cherenkov telescopes. *New Journal of Physics*, 11(5):055005, 2009.
- [41] Gus Sinnis. Air shower detectors in gamma-ray astronomy. *New Journal of Physics*, 11(5):055007, 2009.
- [42] Joel M Weisberg and Joseph H Taylor. Relativistic binary pulsar b1913+ 16: Thirty years of observations and analysis. *arXiv preprint astro-ph/0407149*, 2004.
- [43] BP Abbott, R Abbott, R Adhikari, P Ajith, Bruce Allen, G Allen, RS Amin, SB Anderson, WG Anderson, MA Arain, et al. LIGO: the Laser Interferometer Gravitational-Wave Observatory. *Reports on Progress in Physics*, 72(7):076901, 2009.
- [44] Benjamin P Abbott, Richard Abbott, TD Abbott, MR Abernathy, Fausto Acernese, Kendall Ackley, Carl Adams, Thomas Adams, Paolo Addesso, RX Adhikari, et al. Observation of gravitational waves from a binary black hole merger. *Physical review letters*, 116(6):061102, 2016.
- [45] Christian D Ott. The gravitational-wave signature of core-collapse supernovae. *Classical and Quantum Gravity*, 26(6):063001, 2009.
- [46] Benjamin P Abbott, R Abbott, TD Abbott, MR Abernathy, F Acernese, K Ackley, C Adams, T Adams, P Addesso, RX Adhikari, et al. Properties of the binary black hole merger gw150914. *Physical Review Letters*, 116(24):241102, 2016.
- [47] Abdul H Mroué, Mark A Scheel, Bela Szilagyi, Harald P Pfeiffer, Michael Boyle, Daniel A Hemberger, Lawrence E Kidder, Geoffrey Lovelace, Serguei Ossokine, Nicholas W Taylor, et al. Catalog of 174 binary black hole simulations for gravitational wave astronomy. *Physical Review Letters*, 111(24):241104, 2013.
- [48] Neil J Cornish and Tyson B Littenberg. Bayeswave: Bayesian inference for gravitational wave bursts and instrument glitches. *Classical and Quantum Gravity*, 32(13):135012, 2015.
- [49] Shourov Chatterji, Lindy Blackburn, Gregory Martin, and Erik Katsavounidis. Multiresolution techniques for the detection of gravitational-wave bursts. *Classical and Quantum Gravity*, 21(20):S1809, 2004.
- [50] Luc Blanchet, Thibault Damour, Bala R Iyer, Clifford M Will, and Alan G Wiseman. Gravitational-radiation damping of compact binary systems to second post-newtonian order. *Physical Review Letters*, 74(18):3515, 1995.
- [51] Benjamin P Abbott, Richard Abbott, TD Abbott, F Acernese, K Ackley, C Adams, T Adams, P Addesso, RX Adhikari, VB Adya, et al. Multi-messenger observations of a binary neutron star merger. *Astrophys. J. Lett*, 848(2):L12, 2017.
- [52] Benjamin P Abbott, Rich Abbott, TD Abbott, Fausto Acernese, Kendall Ackley, Carl Adams, Thomas Adams, Paolo Addesso, RX Adhikari, VB Adya, et al. Gw170817: observation of gravitational waves from a binary neutron star inspiral. *Physical Review Letters*, 119(16):161101, 2017.
- [53] A. Goldstein, P. Veres, E. Burns, M. S. Briggs, R. Hamburg, D. Kocevski, C. A. Wilson-Hodge, R. D. Preece, S. Poolakkil, O. J. Roberts, C. M. Hui, V. Connaughton, J. Racusin, A. von Kienlin, T. Dal Canton, N. Christensen, T. Littenberg, K. Siellez, L. Blackburn, J. Broida, E. Bissaldi, W. H. Cleveland, M. H. Gibby, M. M. Giles, R. M. Kippen, S. McBreen, J. McEnery, C. A. Meegan, W. S. Paciesas, and M. Stanbro. An ordinary short gamma-ray burst with extraordinary implications: Fermi-GBM detection of GRB 170817a. *The Astrophysical Journal*, 848(2):L14, oct 2017.
- [54] Z. Hampel-Arias and S. Westerhoff. Gamma-Hadron separation using pairwise compactness method with HAWC. In *34th International Cosmic Ray Conference (ICRC2015)*, volume 34, page 1001, July 2015.

- [55] Peter KF Grieder. Extensive air showers. high energy phenomena and astrophysical aspects. vol. 1. (a tutorial, reference manual and data book) 2nd edition. pages 190–191, 2010.
- [56] A. J. Smith and HAWC Collaboration. HAWC: Design, Operation, Reconstruction and Analysis. In *34th International Cosmic Ray Conference (ICRC2015)*, volume 34, page 966, July 2015.
- [57] AU Abeysekara, A Albert, R Alfaro, C Alvarez, JD Álvarez, R Arceo, JC Arteaga-Velázquez, HA Ayala Solares, AS Barber, N Bautista-Elivar, et al. Observation of the Crab nebula with the HAWC gamma-ray observatory. *The Astrophysical Journal*, 843(1):39, 2017.
- [58] Koichi Kamata and Jun Nishimura. The lateral and the angular structure functions of electron showers. *Progress of Theoretical Physics Supplement*, 6:93–155, 1958.
- [59] The HAWC Collaboration. The Crab Nebula spectrum extending up to 100 tev as seen by the HAWC Gamma-Ray Observatory. *Submitted to The Astrophysical Journal*, 2019. In preparation.
- [60] Joshua Wood. Results from the first one and a half years of the HAWC GRB program. *arXiv preprint arXiv:1801.01437*, 2018.
- [61] Patrick W. Younk, Robert J. Lauer, Giacomo Vianello, J. Patrick Harding, Hugo Alberto Ayala Solares, Hao Zhou, and Michelle Hui for the HAWC Collaboration. A high-level analysis framework for HAWC. *arXiv e-prints*, Aug 2015.
- [62] "hierarchical equal area isolatitude pixelation". <https://healpix.sourceforge.io/>, 2018. [Online; accedida por última vez en Febrero 2019].
- [63] R Atkins, W Benbow, D Berley, E Blaufuss, J Bussons, DG Coyne, RS Delay, T DeYoung, BL Dings, DE Dorfan, et al. Observation of TeV gamma rays from the Crab nebula with Milagro using a new background rejection technique. *The Astrophysical Journal*, 595(2):803, 2003.
- [64] Principal component methods in R: a practical guide. <http://www.sthda.com/english/articles/31-principal-component-methods-in-r-practical-guide/112-pca-principal-component-analysis-essentials/>, 2017. [Online; accedida por última vez en Noviembre 2018].
- [65] Henry F Kaiser. A note on guttman's lower bound for the number of common factors 1. *British Journal of Statistical Psychology*, 14(1):1–2, 1961.
- [66] Ian T Jolliffe. Graphical representation of data using principal components. *Principal component analysis*, pages 78–110, 2002.
- [67] Pedro R Peres-Neto, Donald A Jackson, and Keith M Somers. How many principal components? stopping rules for determining the number of non-trivial axes revisited. *Computational Statistics & Data Analysis*, 49(4):974–997, 2005.
- [68] Francois Chollet. *Deep learning with python*. Manning Publications Co., 2017.
- [69] P Sanmuti and N Puri. Symbolic network analysis-an algebraic formulation. *IEEE Transactions on Circuits and Systems*, 27(8):679–687, 1980.
- [70] David E Rumelhart, Geoffrey E. Hinton, and Ronald J. Williams. Learning representations by back-propagating errors. *Nature*, 323:533–536, 1986.
- [71] "keras: The python deep learning library". <https://keras.io/>, 2019. [Online; accedida por última vez en Marzo 2019].
- [72] AU Abeysekara, A Albert, R Alfaro, C Alvarez, JD Álvarez, R Arceo, JC Arteaga-Velázquez, HA Ayala Solares, AS Barber, B Baughman, et al. The 2hwc hawc observatory gamma-ray catalog. *The Astrophysical Journal*, 843(1):40, 2017.
- [73] I. Martinez-Castellanos and A.J. Smith on behalf of the HAWC Collaboration. LIGO/Virgo G298048: HAWC follow-up. GCN Circular, 2017.
- [74] LAT Data products. <https://fermi.gsfc.nasa.gov/ssc/data/access/lat/>, 2019. [Online; accedida por última vez en Marzo 2019].
- [75] Peter Karl Grieder. *Extensive air showers*. Springer Berlin, 2010.

- [76] JP Norris, TL Cline, UD Desai, and BJ Teegarden. Frequency of fast, narrow γ -ray bursts. *Nature*, 308(5958):434, 1984.
- [77] Chryssa Kouveliotou, Charles A Meegan, Gerald J Fishman, Narayana P Bhat, Michael S Briggs, Thomas M Koshut, William S Paciesas, and Geoffrey N Pendleton. Identification of two classes of gamma-ray bursts. *The Astrophysical Journal*, 413:L101–L104, 1993.
- [78] BP Abbott, R Abbott, TD Abbott, F Acernese, K Ackley, C Adams, T Adams, P Addesso, RX Adhikari, VB Adya, et al. Gravitational waves and gamma-rays from a binary neutron star merger: Gw170817 and grb 170817a. *The Astrophysical Journal Letters*, 848(2):L13, 2017.
- [79] David Eichler, Mario Livio, Tsvi Piran, and David N Schramm. Nucleosynthesis, neutrino bursts and γ -rays from coalescing neutron stars. *Nature*, 340(6229):126, 1989.
- [80] Ramesh Narayan, Bohdan Paczyński, and Tsvi Piran. Gamma-ray bursts as the death throes of massive binary stars. *arXiv preprint astro-ph/9204001*, 1992.
- [81] S. D. Barthelmy, W. H. Baumgartner, J. R. Cummings, E. E. Fenimore, N. Gehrels, H. A. Krimm, A. Y. Lien, C. B. Markwardt, A. Melandri, D. M. Palmer, T. Sakamoto, G. Sato, M. Stamatikos, J. Tueller, and T. N. Ukwatta. GRB 130603B: Swift-BAT refined analysis. *GRB Coordinates Network, Circular Service, No. 14741, #1 (2013)*, 14741, 2013.
- [82] J. Norris, N. Gehrels, S. D. Barthelmy, and T. Sakamoto. GRB 130603B: Swift-BAT spectral lag analysis. *GRB Coordinates Network, Circular Service, No. 14746, #1 (2013)*, 14746, 2013.
- [83] S. Golenetskii, R. Aptekar, D. Frederiks, E. Mazets, V. Pal’Shin, P. Oleynik, M. Ulanov, D. Svinkin, and T. Cline. Konus-wind observation of GRB 130603B. *GRB Coordinates Network, Circular Service, No. 14771, #1 (2013)*, 14771, 2013.
- [84] Albert Einstein. Approximative integration of the field equations of gravitation. *Sitzungsber. Preuss. Akad. Wiss. Berlin (Math. Phys.)*, 1916:688–696, 1916.
- [85] Albert Einstein. Über gravitationswellen. *Sitzungsberichte der Königlich Preußischen Akademie der Wissenschaften (Berlin)*, Seite 154-167., 1918.
- [86] Tev gamma-ray source catalog. <http://tevcat.uchicago.edu/>, 2019. [Online; accedida por última vez en Marzo 2019].
- [87] Jim Braun. Event reconstruction in HAWC. Presentación Interna de HAWC, 2016.
- [88] R Atkins, W Benbow, D Berley, ML Chen, DG Coyne, BL Dingus, DE Dorfan, RW Ellsworth, D Evans, A Falcone, et al. The high-energy gamma-ray fluence and energy spectrum of grb 970417a from observations with milagro. *The Astrophysical Journal*, 583(2):824, 2003.
- [89] AA Abdo, BT Allen, D Berley, E Blaufuss, S Casanova, BL Dingus, RW Ellsworth, MM Gonzalez, JA Goodman, E Hays, et al. Milagro constraints on very high energy emission from short-duration gamma-ray bursts. *The Astrophysical Journal*, 666(1):361, 2007.
- [90] R Atkins, W Benbow, D Berley, E Blaufuss, DG Coyne, T DeYoung, BL Dingus, DE Dorfan, RW Ellsworth, L Fleysher, et al. Constraints on very high energy gamma-ray emission from gamma-ray bursts. *The Astrophysical Journal*, 630(2):996, 2005.
- [91] R Atkins, W Benbow, D Berley, E Blaufuss, J Bussons, DG Coyne, T DeYoung, BL Dingus, DE Dorfan, RW Ellsworth, et al. Limits on very high energy emission from gamma-ray bursts with the milagro observatory. *The Astrophysical Journal Letters*, 604(1):L25, 2004.
- [92] AU Abeysekara, JA Aguilar, S Aguilar, R Alfaro, E Almaraz, C Álvarez, J de D Álvarez-Romero, M Álvarez, R Arceo, JC Arteaga-Velázquez, et al. On the sensitivity of the hawc observatory to gamma-ray bursts. *Astroparticle Physics*, 35(10):641–650, 2012.
- [93] LIGO Scientific Collaboration, Virgo Collaboration, et al. GWTC-1: a gravitational-wave transient catalog of compact binary mergers observed by LIGO and Virgo during the first and second observing runs. *arXiv preprint arXiv:1811.12907*, 2018.