Instituto Politécnico Nacional

Centro de Investigación en Computación

TESIS

Aprendizaje automático para la clasificación de datos de colisiones de altas energías del experimento CMS-LHC

para obtener el grado de:

Maestría en Ciencias de la Computación

presenta:

Fis. Saksevul Arias Santiz

directores de tesis:

Dr. René Luna García
Dr. Hermes León Vargas

Ciudad de México Agosto 2022

SIP-13
REP 2017

INSTITUTO POLITÉCNICO NACIONAL
SECRETARIA DE INVESTIGACIÓN Y POSGRADO

ACTA DE REGISTRO DE TEMA DE TESIS
Y DESIGNACIÓN DE DIRECTOR DE TESIS

Ciudad de México, a 02 de marzo del 2021

El Colegio de Profesores de Posgrado del **Centro de Investigación en Computación** en su Sesión
(Unidad Académica)

Ordinaria No 12 celebrada el día 21 del mes diciembre de 2020, conoció la solicitud presentada por el (la) alumno (a):

Apellido Paterno:	ARIAS	Apellido Materno:	SANTIZ	Nombre (s):	SAKSEVUL

Número de registro: B 2 0 0 4 2 1

del Programa Académico de Posgrado: **Maestría en Ciencias de la Computación**

Referente al registro de su tema de tesis; acordando lo siguiente:

1.- Se designa al aspirante el tema de tesis titulado:

"Aprendizaje automático para la clasificación de datos de colisiones de altas energías del experimento CMS-LHC"

Objetivo general del trabajo de tesis:

Desarrollar la metodología completa que abarca desde la obtención de los datos a utilizar, hasta su implementación en una red neuronal para su clasificación. Pasando por el desarrollo, entrenamiento e implementación recursiva de esta red.

2.- Se designa como Directores de Tesis a los profesores:

Director: **Dr. René Luna García** 2° Director: **Dr. Hermes León Vargas**

No aplica: ☐

3.- El Trabajo de investigación base para el desarrollo de la tesis será elaborado por el alumno en:

Centro de Investigación en Computación

que cuenta con los recursos e infraestructura necesarios.

4.- El interesado deberá asistir a los seminarios desarrollados en el área de adscripción del trabajo desde la fecha en que se suscribe la presente, hasta la aprobación de la versión completa de la tesis por parte de la Comisión Revisora correspondiente.

Director de Tesis
Dr. René Luna García

2º Director de Tesis
Dr. Hermes León Vargas

Aspirante
Saksevul Arias Santiz

Presidente del Colegio
Dr. Marco Antonio Moreno Ibarra

INSTITUTO POLITECNICO NACIONAL
CENTRO DE INVESTIGACION
EN COMPUTACION
DIRECCION

SIP-14
REP 2017

INSTITUTO POLITÉCNICO NACIONAL

SECRETARÍA DE INVESTIGACIÓN Y POSGRADO

ACTA DE REVISIÓN DE TESIS

En la Ciudad de México siendo las 12:45 horas del día 22 del mes de junio del 2022 se reunieron los miembros de la Comisión Revisora de la Tesis, designada por el Colegio de Profesores de Posgrado de: Centro de Investigación en Computación para examinar la tesis titulada:

"Aprendizaje automático para la clasificación de datos de colisiones de altas energías del experimento CMS-LHC"

del (la) alumno (a):

Apellido Paterno:	ARIAS	Apellido Materno:	SANTIZ	Nombre (s):	SAKSEVUL

Número de registro: B 2 0 0 4 2 1

Aspirante del Programa Académico de Posgrado: **Maestría en Ciencias de la Computación**

Una vez que se realizó un análisis de similitud de texto, utilizando el software antiplagio, se encontró que el trabajo de tesis tiene 13 % de similitud. **Se adjunta reporte de software utilizado**.

Después que esta Comisión revisó exhaustivamente el contenido, estructura, intención y ubicación de los textos de la tesis identificados como coincidentes con otros documentos, concluyó que en el presente trabajo **SI** ☐ **NO** ☒ **SE CONSTITUYE UN POSIBLE PLAGIO.**

JUSTIFICACIÓN DE LA CONCLUSIÓN: *(Por ejemplo, el % de similitud se localiza en metodologías adecuadamente referidas a fuente original)*

Esta dentro del valor aceptable

****Es responsabilidad del alumno como autor de la tesis la verificación antiplagio, y del Director o Directores de tesis el análisis del % de similitud para establecer el riesgo o la existencia de un posible plagio.**

Finalmente y posterior a la lectura, revisión individual, así como el análisis e intercambio de opiniones, los miembros de la Comisión manifestaron **APROBAR** ☒ **SUSPENDER** ☐ **NO APROBAR** ☐ la tesis por **UNANIMIDAD** ☒ o **MAYORÍA** ☐ en virtud de los motivos siguientes:

Cumple con todos los requisitos preestablecidos

COMISIÓN REVISORA DE TESIS

Dr. René Luna García
Director de Tesis

Dr. Edgardo Manuel Felipe Riverón

M. en C. Germán Téllez Castillo

Dr. Hermes León Vargas
2° Director de Tesis

Dr. Carlos Fernando Aguilar Ibañez

Dr. Gilberto Lorenzo Martínez Luna

Dr. Francisco Hiram Calvo Castro
PRESIDENTE DEL COLEGIO DE PROFESORES

INSTITUTO POLITÉCNICO NACIONAL
SECRETARÍA DE INVESTIGACIÓN Y POSGRADO
DIRECCIÓN
IPN-CIC

Página 1 de 1

INSTITUTO POLITÉCNICO NACIONAL
SECRETARÍA DE INVESTIGACIÓN Y POSGRADO

CARTA DE AUTORIZACIÓN DE USO DE OBRA PARA DIFUSIÓN

En la Ciudad de México el día 08 del mes de agosto del año 2022, el que suscribe, Saksevul Arias Santiz alumno del programa Maestría en Ciencias de la Computación con número de registro B200421, adscrito al Centro de Investigación en Computación, manifiesta que es autor intelectual del presente trabajo de tesis bajo la dirección del Dr. René Luna García y del Dr. Hermes León Vargas y cede los derechos del trabajo intitulado *Aprendizaje automático para la clasificación de datos de colisiones de altas energías del experimento CMS-LHC*, al Instituto Politécnico Nacional, para su difusión con fines académicos y de investigación.

Los usuarios de la información no deben reproducir el contenido textual, gráficas o datos del trabajo sin el permiso expresado del autor y/o directores. Este puede ser obtenido escribiendo a las siguientes direcciones de correo: saksevul@cic.ipn.mx, lunar@cic.unam.mx y hleonvar@fisica.unam.mx. Si el permiso se otorga, al usuario deberá dar agradecimiento correspondiente y citar la fuente de este.

Saksevul Arias Santiz

Resumen

Esta tesis se centra en el problema de la clasificación de eventos en física de altas energías. La metodología implementada es la conocida como *End-to-End*, la cual se basada en redes neuronales artificiales aplicadas a imágenes que son reconstruidas a partir de las señales electrónicas del detector. Su principal ventaja es que el clasificador tiene acceso a la mayor cantidad de información recolectada por el detector. Se utilizan datos abiertos del experimento CMS del LHC.

Los eventos en los cuales se enfoca este trabajo son los DiJet y MultiJet, los cuales son caracterizados por las propiedades cinemáticas de los jets en el evento. Previo a la clasificación de los eventos, se propone una segmentación de los jets. El segmentador utilizado se basa en la red `ResNet-101` y la clasificación de eventos es hecha por la red `ResJet`, la cual es una modificación de las Redes Residuales que trabaja con las características de los eventos y los jets contenidos en estos.

Para evaluar el funcionamiento del clasificador (`ResJet`), se compara su desempeño con la red utilizada en el estado del arte de la metodología *End-to-End*, la red `ResNet-34`. La comparación es hecha utilizando imágenes segmentadas y no-segmentadas. Los resultados obtenidos reflejan una mejora al clasificar eventos no-segemtados con la `ResJet`, en comparación con la `ResNet-34`. La mejora corresponde a un incremento de 0.022 y de 0.023 puntos para el *precision* y *recall*, respectivamente. Por otra parte, el uso de datos segmentados implicó una mejoría de 0.05 y 0.04 puntos en *precision* y *recall* (respectivamente), al utilizar la red `ResNet-34`.

Abstract

This thesis focuses the problem of event classification in high energy physics. The implemented methodology is known as End-to-End, which is based on artificial neural networks applied to images that are reconstructed from the electronic signals of the detector. Its main advantage is that the classifier has access to the largest amount of information collected by the detector. Open data from the CMS experiment is used.

The events on which this work focuses are the DiJet and MultiJet, which are characterized by the kinematic properties of the jets in the event. Prior to the classification of the events, a segmentation of the jets is proposed. The segmenter used is based on the `ResNet-101` network and the classification of events is done by the `ResJet` network, which is a modification of the Residual Networks that works with the characteristics of the events and the jets contained in these.

To evaluate the performance of the classifier (`ResJet`), its performance is compared with the network used in the state-of-the-art of the End-to-End methodology, the `ResNet-34` network. The comparison is made using segmented and non-segmented images. The results obtained reflect an improvement when classifying non-segmented events with the `ResJet`, compared to the `ResNet-34`. The improvement corresponds to an increase of 0.022 and 0.023 points for accuracy and recall, respectively. On the other hand, the use of segmented data implied an improvement of 0.05 and 0.04 points in accuracy and recall (respectively), when using the `ResNet-34` network.

Agradecimientos

Al CERN por haber publicado sus datos para el libre acceso, así debería ser la ciencia.

Al Instituto Politécnico Nacional, por permitirme continuar mi formación académica, en particular al Centro de Investigación en Computación.

Al Dr. René Luna García, por su continua enseñanza desde el primer semestre, por la constante presión, por toda su dedicación para lograr esta meta. Más que un asesor, lo considero un mentor.

A la Dr. Hermes León Vargas, por su apoyo y asesoría en la investigación, por sus aportaciones y sugerencias en este trabajo.

A mi comisión revisora de tesis, el Dr. Edgardo Manuel Felipe Riverón, al M. en C. Germán Tellez Castillo, al Dr. Gilberto Lorenzo Martinez Luna y al Dr. Carlos Fernando Aguilar Ibañez, por la información transmitida y el conocimiento proporcionado para la realización y revisión de este trabajo.

Al Consejo Nacional de Ciencia y Tecnología (CONACyT), por la beca otorgada para realizar mis estudios de maestría.

¡Muchas gracias!

Glosario

AOD (*Analysis Object Data*): Datos de Objetos de Análisis, un formato de datos utilizado para almacenar información referente a colisiones de partículas.

CERN (*Conseil Européen pour la Recherche Nucléaire*): Organización Europea para la Investigación Nuclear.

Clusters: Agrupaciones o amontonamientos, por ejemplo de energía o de partículas.

CMS (*Compact Muon Solenoid*): Solenoide Compacto de Muones, el cual es un detector de partículas multipropósito y se encuentra en el LHC.

CNN (Red Neuronal Convolucional): Red neuronal artificial especializada en la extracción de características.

ECAL (Calorímetro Electromagnético): Parte del detector CMS que detecta la energía de los electrones y fotones.

Evento(s): Esta palabra cuyo significado depende del contexto; ver secciones 2.2.2, 2.2.4.

FastJet: Es una paquetería computacional el cual se dedica a la reconstrucción de jets.

HCAL (Calorímetro Hadrónico): Parte del detector CMS que detecta la energía de los hadrones.

IPN (Instituto Politécnoco Nacional).

Jet: Conjunto de partículas que viajan en una misma dirección y sentido. La característica principal de estas partículas es que han sido generadas a partir de un mismo partón (*quark* o gluón).

LHC (*Large Hadron Collider*): Gran Colisionador de Hadrones, es un acelerador de partículas encontrado en el CERN.

Macro (Macro-instrucción): En términos prácticos es una instrucción compleja, formada por otras instrucciones más sencillas, la cual es interpretada y ejecutada mediante algún lenguaje de programación.

MV (Máquina Virtual): En la explicación más banal, es una computadora simulada dentro de otra computadora.

PF (ParticleFlow): Algoritmo de reconstrucción de partículas a partir de la información recolectada por un detector de partículas.

Pseudo-rapidez (η): Es una coordenada espacial utilizada por CMS, proviene de la transformación de la coordenada θ del sistema coordenado esférico.

ResNet (Residual Network): Red neuronal que se basa en bloques residuales, los cuales son propuestos para atacar el problema de degradación en las redes neuronales profundas.

Root: Entorno de trabajo basado en C++ y desarrollado para el análisis de datos.

Índice general

Índice de figuras

Capítulo 1

Introducción

Los componentes de la materia conocida en el universo son los átomos, que a su vez están conformados por electrones y nucleones. Los primeros forman parte de la familia de los leptones, mientras que los segundos son conocidos como hadrones (partículas formadas por quarks). En proporción a su masa, los electrones son aproximadamente un 0.05 % la masa de los nucleones. Esto quiere decir que si somos capaces de entender el comportamiento los quarks (y por ende a los gluones -que son las partículas mediante las cuales interaccionan-), entonces seríamos capaces de comprender cerca del 99.95 % de la materia conocida en el universo.

Uno de los resultados relevantes de la Cromodinámica Cuántica es que los quarks y gluones no pueden existir libremente debido a la restricción llamada Carga de Color, la cual debe ser siempre neutra. Es por esto que se requieren métodos indirectos para estudiar a los componentes fundamentales de la materia hadrónica (los partones -quarks y gluones-).

De manera experimental, los colisionadores de hadrones brindan la oportunidad de estudiar la información proveniente de las interacciones entre partones. Sin embargo, la cantidad de eventos[1] que se pueden almacenar es cerca de una millonésima parte de la cantidad total de eventos que se generan. Por lo tanto, es necesario conocer cuales de estos son los más relevantes. De manera tradicional, para lograr este cometido es

[1] A la información (nuevas partículas generadas y sus propiedades cinemáticas) resultante de una colisión de partículas se le conoce como "evento". Por esta razón, en el contexto de la física de partículas no se habla de colisiones *per se* sino de sus eventos asociados.

necesario pasar por un procesamiento complejo para seleccionar si un evento es lo suficientemente relevante para ser almacenado.

En el caso de los detectores de partículas, los métodos tradicionales para realizar esta selección de eventos consisten en comenzar a filtrar de manera muy metodológica las señales electrónicas, hasta llegar a tener información de partículas reconstruidas. Este factor tiene un inconveniente: la pérdida de la información, pues se utiliza continuamente "todo lo que se conoce", lo que potencialmente hace que se obvie información desconocida y limita la búsqueda de nuevas propiedades físicas dentro de cada colisión.

1.1. Problema a resolver

En las colisiones de partículas a altas energías, los eventos se clasifican en función de las partículas generadas, así como de sus propiedades cinemáticas. Debido a la naturaleza de la detección de las partículas, es necesario realizar una reconstrucción previa de cada una de estas, utilizando la información recolectada por las distintas partes del detector (los sub-detectores).

Este trabajo ataca el problema de la clasificación de eventos de una manera distinta, mediante técnicas de Inteligencia Artificial aplicadas a las señales "crudas" del detector. Esto es, clasificar los eventos sin realizar una reconstrucción previa de las partículas resultantes de la colisión.

Para realizar la clasificación se utiliza un método basado en imágenes, las cuales están conformadas por las señales obtenidas por las distintas partes del detector. De este modo, un evento estará representado por una imagen RGB cuyos planos contienen información proveniente de distintos sub-detectores. Posteriormente, se utilizan estas imágenes para entrenar una red neuronal artificial, la cual se encarga de clasificar los eventos.

1.2. Justificación

Gracias a los trabajos correspondientes a la metodología *End-to-End*[2] citados y discutidos en el Estado del Arte (Capítulo 3 de este trabajo), se sabe que es posible realizar clasificaciones referentes a eventos, mediante la representación por imágenes de las señales de los detectores.

La metodología *End-toEnd* brinda mejoras técnicas desde el punto de vista computacional. La primera de ellas es que la complejidad computacional se ve reducida considerablemente. Esto se debe principalmente a que se evita la reconstrucción de las partículas generadas en cada evento. Por otra parte, no realiza una discriminación de la información recolectada por el detector, esto es, no realiza ningún intento por eliminar el "ruido" de las señales. Desde el punto de vista físico, esto brinda una ventana para el uso de posibles propiedades físicas que son desconocidas por los científicos pero que pueden ser utilizables por las herramientas de Inteligencia Artificial.

1.3. Hipótesis

Se espera que la base de datos considerada en este trabajo (datos reales) pueda ser utilizada pese a los inconvenientes que muestra al compararla con los datos utilizados en el estado del arte (datos simulados). Mediante la representación por imágenes de las señales obtenidas por el detector se espera entrenar a una red neuronal desarrollada específicamente para clasificar eventos. Se utilizarán como patrones clave de la clasificación a unos entes físicos llamados *Jets*. Los cuales se definen como "chorros" colimados de partículas.

Por otra parte, la segmentación de los Jets que se propone, previo al proceso de clasificación de eventos, se espera que conduzca a una mejora de los resultado de la clasificación al compararla con los resultado que se obtendrían al utilizar las técnicas consideradas en el estado del arte.

[2]Metodología en la cual se utiliza directamente la información de las señales crudas de los detectores.

1.4. Objetivos

1.4.1. Objetivo general

Clasificar eventos mediante técnicas basadas en Inteligencia Artificial y Análisis de Imágenes.

1.4.2. Objetivos particulares

- Familiarizarse con el entorno de trabajo de CMS
- Descargar la base de datos
- Convertir los datos numéricos en imágenes
- Evaluar distintas redes neuronales artificiales para realizar una segmentación de patrones
- Evaluar el desempeño de la segmentación previa
- Desarrollar una Red Neuronal Artificial para clasificar los eventos utilizando la información segmentada
- Evaluar el desempeño de la red propuesta en el paso anterior

1.5. Medios utilizados

Los recursos computacionales utilizados consisten en una computadora de escritorio con procesador Intel Core i7-11700M, tarjeta gráfica Mesa Intel Graphics (RKLGT1), memoria RAM de 32 GB y SSD de 480 GB; cuyo sistema operativo es Ubuntu Server 20.04 en la versión de 64 bits. Adicionalmente se cuenta con un disco duro externo de 4TB de almacenamiento en el cual se alojan los datos de física de partículas utilizados.

1.6. Apoyos y contribuciones

1.6.1. Apoyos

La base de datos utilizada corresponde a datos de colisiones reales llevadas a cabo en el año 2011 por el Gran Colisionador de Hadrones en el CERN. Consiste en los datos generados mediante el experimento CMS (Compact Muon Solenoid), los cuales son accesibles al público en general gracias a la iniciativa de datos abiertos del CERN.

1.6.2. Contribuciones

La contribución de este trabajo consiste en desarrollar una metodología computacional para la clasificación de eventos reales de física de partículas. Adicionalmente, se propone una red neuronal convolucional desarrollada específicamente para realizar la clasificación de eventos. De esta manera se complementa al estado del arte, en donde se trabaja con datos simulados y se utilizan redes neuronales previamente desarrolladas y entrenadas.

El software desarrollado puede ser encontrado en el siguiente repositorio de GitHub: `https://github.com/Saksevul/T-MCC`

1.7. Organización del documento

El presente documento comienza en este capítulo introductorio dando un panorama general del trabajo, sin entrar en discusiones específicas. En el capítulo 2 se desarrollan los conceptos necesarios para dominar la jerga y los conceptos técnicos relacionados a este trabajo. El capítulo 3 consiste en una descripción de los trabajos correspondientes al estado del arte de este trabajo. En el capítulo 4 se describe a detalle el procedimiento utilizado para solventar el problema de clasificación descrito al inicio de este capítulo. El capítulo 5 está reservado para dar a conocer los resultados obtenidos, así como su correspondiente evaluación en función del estado del arte. Finalmente, en el capítulo 6 se dan las conclusiones finales, así como una breve

descripción de los futuros trabajos que a realizar.

Capítulo 2

Conceptos básicos y definiciones

2.1. Inteligencia Artificial

Las cualidades de la Inteligencia Artificial (IA) van desde poder devolver una respuesta partiendo de una entrada, hasta la capacidad de aprender y discernir, pasando por la búsqueda de patrones y la resolución de problemas basados en lógica [5]. En términos coloquiales, se dice que la inteligencia se basa en las técnicas y estrategias computacionales y matemáticas para imitar, y en el mejor de los casos superar, la capacidad de la mente humana para resolver problemas y tomar decisiones.

2.1.1. Aprendizaje Automático

El aprendizaje automático se considera una subconjunto de la IA. Se puede entender al aprendizaje como la capacidad de un ente de aumentar su conocimiento para mejorar su comportamiento, y así, obtener mejores resultado durante la interacción con su entorno. Este aprendizaje normalmente ocurre mediante la selección de las características relevantes de un objeto (o evento) y la comparación con un caso similar conocido. En este sentido, el aprendizaje se da cuando las diferencias en la comparación son significativas y el modelo que se tiene de dicho objeto (o evento) debe de adaptarse [6].

Dentro de ésta categoría de IA existen dos grandes ramas: el aprendizaje automá-

tico supervisado y el no supervisado. Para éste trabajo se contempla únicamente el supervisado, en específico, el supervisado de clasificación.

En el algoritmo supervisado de clasificación lo que se espera es que este sea capaz de discernir a qué conjunto corresponde cada elemento del grupo de estudio. Para hacer esto, busca patrones en los datos de entrenamiento y los clasifica en grupos, posteriormente clasifica los nuevos datos en dichos grupos.

Aprendizaje supervisado

Este tipo de aprendizaje es el más familiar a nuestra intuición. Pues, se tienen conjuntos de información etiquetadas y se necesita a un ente que sea capaz de extraer las características relevantes de estos conjuntos para hacer predicciones sobre nuevos conjuntos de datos.

Otra forma de entender el aprendizaje supervisado es que, en este caso, las entradas y las salidas son conocidas. Es decir, el método de aprendizaje consiste en entrenar los algoritmos con ejemplos conocidos, a sabiendas de cual es el resultado esperado. De esta manera, los nuevos ejemplos los clasificará en función de lo aprendido previamente [7].

Algunos de los algoritmos más conocidos en este tipo de aprendizaje son: Regresiones Lineales, K Nearest Neighbours, Máquinas de Soporte Vectorial, Redes Neuronales Artificiales, Naive Bayes, Árboles de Decisión, entre otros.

Aprendizaje no supervisado

El aprendizaje no supervisado se puede entender como el inverso del aprendizaje supervisado. En este sentido, los datos obtenidos no se encuentran etiquetados. Esto significa que el algoritmo empleado ha de ser capaz de encontrar estructuras en los datos, de tal modo que se pueda obtener distintos conglomerados de datos [8].

En otras palabras, a diferencia del aprendizaje supervisado, el no supervisado "clasifica" (mejor dicho, conglomera) los datos en función de las características que el algoritmo considere más relevantes. Estas características pueden ser de cualquier tipo, inclusive alguna que no sean comprensibles por el entendimiento humano [7].

2.1.2. Modelos de aprendizaje automático

Redes neuronales artificiales

Tal como su nombre sugiere, éste modelo de Aprendizaje Automático trata de replicar el comportamiento cerebral, en donde millones de neuronas intercambian mensajes entre sí. En la actualidad este modelo está en auge por el potencial cognitivo que tiene. En especial, en la compleja tarea del reconocimiento de imágenes o vídeos es en donde destaca respecto a otros modelos.

En términos matemáticos, se tiene que las Redes neuronales (artificiales) toman como entrada un vector formado por los datos de entrada. Posteriormente, realiza productos tensoriales a estos datos, y el resultado es procesado por una función de activación (generalmente no lineal). Finalmente, este proceso se repite a través de todas las capas de la red. Esto resulta en que los datos de salida son un transformación no lineal de los datos de entrada [7].

Redes neuronales convolucionales

Las Redes Neuronales Convolucionales (CNN por sus siglas en inglés) son consideradas como un algoritmo efectivo de reconocimiento que es utilizado en distintas aplicaciones y en especial en la tareas que tengan que ver con reconocimiento de patrones y procesamiento de imágenes. Entre sus principales ventajas, se encuentra su menor cantidad de parámetros (en comparación con la red Perceptrón) y su mayor adaptabilidad [9].

En términos prácticos, el algoritmo de una CNN consiste en un perceptrón multicapa diseñado espacialmente para identificar información en arreglos bidimensionales. El trabajo de este tipo de redes consiste en extraer información y reducir el tamaño de los arreglos bidimensionales de entrada. Esta es la razón por la cual es utilizada ampliamente en los trabajos en donde las imágenes están involucradas.

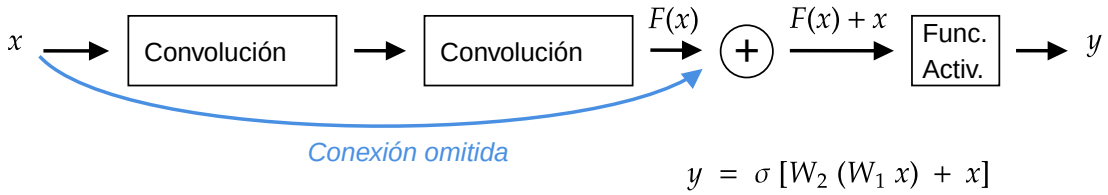


Figura 2.1: Bloque residual.

Redes Neuronales Residuales

Dentro del conjunto de las redes neuronales convolucionales, existe un tipo llamado Residual [10] debido a su topología específica.

El objetivo principal de este tipo de redes es atacar el problema de "degradación": cuando la profundidad de las redes aumenta, el *accuracy* se satura y se degrada rápidamente; este problema no es causado por *overfiting* y agregar más capas solo incrementa el error de entrenamiento.

La idea detrás de una red residual consiste en entregar una copia de la información original a la red después de cierto número de pasos (ver figura 2.1).

Dicho de otro modo, el mapeo de fondo después de pasar por el bloque residual es $H(x) = F(x) + x$. Este tipo de arreglo, en donde se "salta" una o más capas, es conocido como conexión omitida (*skip conection*) [11]. En la imagen anterior se mapea una conexión de identidad y es sumada a la salida de las capas intermedias.

Los beneficios de utilizar este tipo de bloques en las redes es[10]:

- A una misma profundidad, resulta más fácil entrenar que sus contra partes sin conexión omitida.
- A mayor profundidad, el *accuracy* no tiende a empeorar.

Una de las variantes más famosas de este tipo de redes es la `ResNet-34`. Esta red fue desarrollada para competir en el ILSVRC[1] del 2015, en donde ganó el primer lugar. Esta red consta de 16 bloques convolucionales, 1 capa convolucional de entrada y una capa de salida. En total esta red cuenta con 12.3 millones de parámetros.

[1]ImageNet Large Scale Visual Recognition Challenge.

Desempeño de los clasificadores

La primer métrica (aunque no las más importante) que se considera en el proceso de clasificación de este trabajo es el *Accuracy*. Dado que la clasificación se realizará en términos de 3 clases, entonces la definición utilizada de esta métrica es: la fracción total de la clasificación correcta. En términos matemáticos:

$$Accuracy = \frac{\text{Clasificacion} \quad \text{correcta}}{\text{Todas} \quad \text{las} \quad \text{clasificaciones}}$$

En segundo lugar, se tiene al *Precision* (también conocido como valor predictivo positivo) se define como el cociente de los verdaderos positivos (TP) entre la suma de los verdaderos positivos más los falsos positivos (FP). Es decir,

$$Precision = \frac{\text{TP}}{\text{TP} + \text{FP}}$$

Finalmente, el *Recall* (en ocaciones llamado sensitividad) se denota como el cociente de los verdaderos positivos entre la suma de los verdaderos positivos más los falsos negativos (FN).

$$Recall = \frac{\text{TP}}{\text{TP} + \text{FN}}$$

Por otro lado, se tiene la matriz de confusión, la cual permite visualizar el desempeño de un algoritmo basado en aprendizaje supervisado. Las columnas de la matriz representan los resultados de las clases predichas, mientras que los renglones representan las clases reales. Esto permite ver que clases se están confundiendo.

		Predicho		
		Clase 1	Clase 2	Clase 3
Real	Clase 1	100	0	10
	Clase 2	10	80	10
	Clase 3	30	0	70

Tabla 2.1: Matriz de confusión.

2.2. Colisiones de partículas elementales

Uno de los tipos de colisiones más comunes llevadas a cabo en los colisionadores de partículas es el de protón-protón. Los protones forman parte de la familia de los hadrones, lo que significa que están conformados por *quarks* y es por este motivo que, en una colisión a alta energía[2] entre protones, se estudian las interacciones entre sus constituyentes, los *quarks* y los gluones.

Por lo tanto, el contenido de *quarks* en un protón está dado por los *quarks* de valencia (u y d), y el mar de *quarks* y gluones. Por lo tanto, en una colisión de protones, cualquier combinación de *quarks* o gluones puede contribuir al proceso de dispersión fuerte. La probabilidad de que un partón[3] específico, con una fracción de momento del protón, participe en un proceso fuerte es conocido como función de distribución partónica y se mide usando datos de experimentos de dispersión inelástica profunda [12]. La energía de cada partón (*quark* o gluón) es una fracción desconocida de la energía del protón del que forma parte.

Una representación ilustrativa de este fenómeno la podemos encontrar en la figura 2.2, donde se muestra a dos protones a punto de colisionar. Notemos que dentro de ellos está representado el mar de *quarks* y gluones.

Si bien es cierto que dentro de cada protón está un mar de *quarks* y gluones, en realidad los que le dan la identidad de protón a cada protón son los llamados *quarks* de valencia.

2.2.1. CERN

El Consejo Europeo de Investigación Nuclear (CERN por sus siglas en francés) es el mayor centro para la investigación en física de partículas. Fue concebido en la década de los 50's y desde entonces nuestro entendimiento de la materia va mucho más allá del núcleo y la principal área de investigación en el CERN es la física de

[2]En este trabajo consideraremos del orden de GeV para la energía en el centro de masa en las colisiones, pues a esto nos restringe los datos abiertos.

[3]De la misma manera que un nucleón puede ser un protón o un neutrón, llamamos partón a un *quark* o a un gluón.

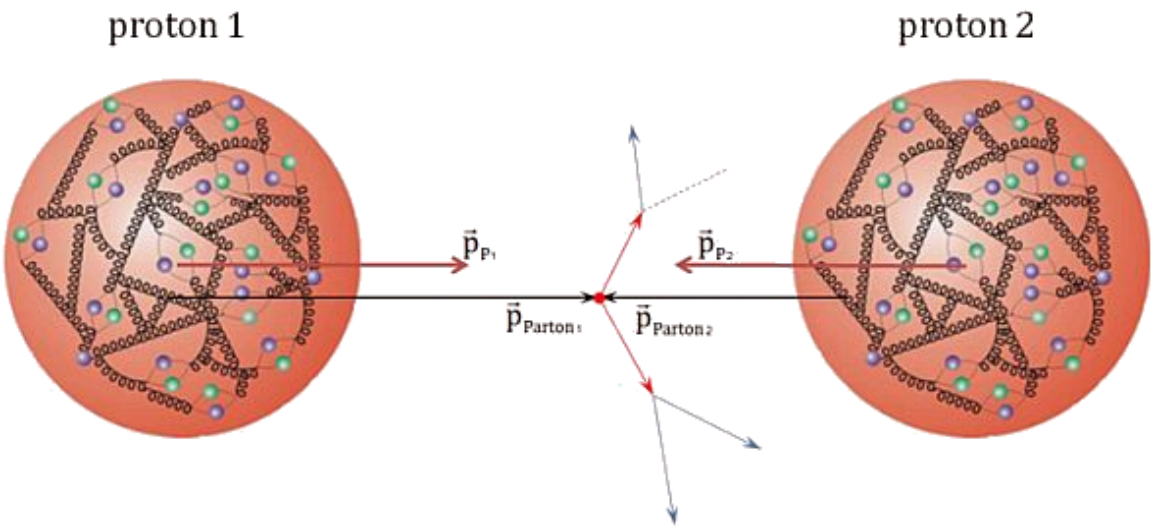


Figura 2.2: Representación de la interacción entre los constituyentes (*quarks* y gluones) de dos protones colisionando. El momento del protón 1 es $\vec{p}_{P_1}$, el momento del protón 2 es $\vec{p}_{P_2}$, el momento del partón 1 es $\vec{p}_{parton1}$ y el momento del partón 2 es $\vec{p}_{parton2}$ [1].

partículas. Es por esto que generalmente es conocido como Laboratorio Europeo para Física de Partículas [13]. En términos generales, la misión principal de CERN es conocer de qué está hecho y cómo funciona el Cosmos [14].

El CERN es una organización europea de investigación que se encuentra en el cantón de Ginebra en la frontera Franco-Suiza y opera el más grande laboratorio de física de partículas en el mundo.

2.2.2. LHC

Para estudiar la física de partículas en el CERN, se tiene al Gran Colisionador de Hadrones (LHC), un acelerador y colisionador de partículas que consta de dos anillos de 27 km de circunferencia y está construido dentro de un túnel. Las motivaciones para la construcción de este colisionador son principalmente el descubrimiento del bosón de Higgs y el estudio de eventos[4] con una energía de colisión en el centro de masa de hasta 14 TeV. El LHC está diseñado como un colisionador protón-protón [15].

Para estudiar las colisiones producidas por el LHC, el CERN cuenta con 4 detec-

[4]En términos experimentales se llama **evento** a la información de todos los productos de una colisión de partículas.

tores principales: ALICE, ATLAS, CMS y LHCb. Los dos detectores más grandes, ATLAS y CMS son detectores de partículas diseñados para ver una amplia gama de partículas y fenómenos producidos en colisiones de alta energía[16].

2.2.3. CMS

El LHC es capaz de colisionar grupos de protones a una velocidad cercana a la de la luz, 40 millones de veces por segundo. Cuando esto sucede, parte de la energía de la colisión se convierte en masa y en partículas de vida corta, antes no observadas, que podrían dar pistas sobre cómo se comporta la naturaleza en un nivel fundamental.

El Solenoide Compacto de Muones (CMS) es un detector de propósito general diseñado para estudiar la física de colisiones protón-protón con una energía en el centro de masa de 14 TeV. Este detector está diseñado para medir la energía, el momento y la trayectoria de fotones, electrones, muones y otras partículas cargadas con una alta precisión [17].

En términos generales el CMS es un detector que consiste en dos calorímetros [18]. Las partículas electromagnéticas (electrones y fotones) son detenidas y medidas en el primero; las partículas hadrónicas son medidas en ambos calorímetros y son detenidas en el segundo. Un sistema de *tracking* mide las trayectorias de todas las partículas cargadas. Y finalmente, el sistema más externo detecta las partículas cargadas que cruzan ambos calorímetros, esto es, muones.

Convención del sistema coordenado

Para describir los eventos detectados y reconstruidos por CMS, se utiliza un sistema coordenado (ver figura 2.3) que tiene el origen centrado en el punto de colisión nominal, dentro del experimento [19]. El eje x apuntando paralelo a la horizontal y en dirección hacia el centro del LHC. El eje y apuntando en dirección a la vertical y el eje z en dirección del haz, apuntando en dirección hacia las montañas *Jura*. El ángulo azimutal ϕ es medido desde el eje x y se encuentra contenido en el plano x-y. El ángulo polar θ se mide a partir del eje z.

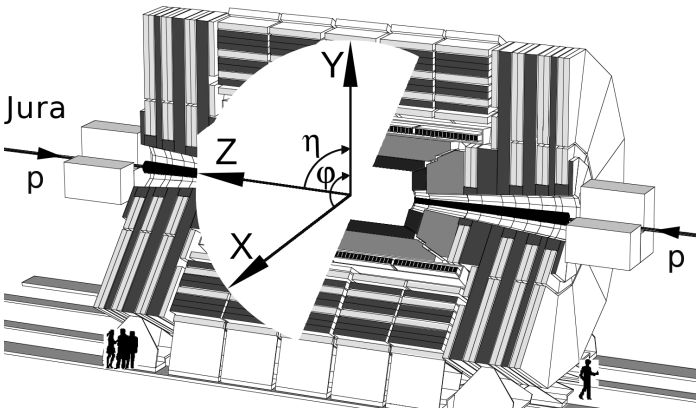


Figura 2.3: Sistema coordenado en el experimento CMS [2].

A partir del momento es posible obtener el **momento transverso** ($\mathbf{p_T}$), el cual se define como la cantidad de momento que tiene cualquier partícula resultante en dirección perpendicular al eje de colisión. Es decir, la proyección del momento en el plano x-y.

Por otra parte, es posible obtener una coordenada espacial llamada *pseudo-rapidez*, η, la cual se define como un ángulo de movimiento, para alguna partícula resultante, con respecto al eje de colisión [20]:

$$\eta \equiv -\ln\left[\tan\left(\frac{\theta}{2}\right)\right]$$

donde θ es el ángulo polar del sistema coordenado esférico.

De manera equivalente, es posible escribir esta ecuación en términos del momento total ($\vec{p}$) de la partícula generada:

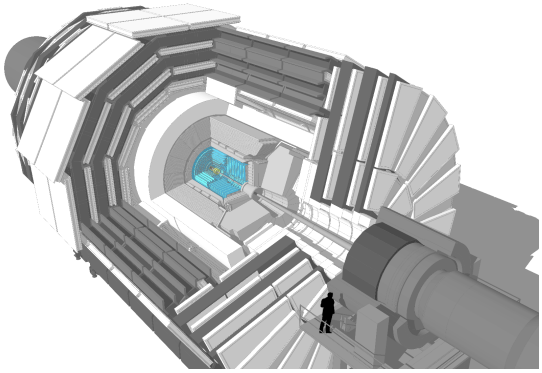
$$\eta = \frac{1}{2}\ln\left(\frac{p_{\parallel}}{|\vec{p}|}\right)$$

donde $p_{\parallel}$ es la proyección del momento total de la partícula en dirección paralela al eje de colisión.

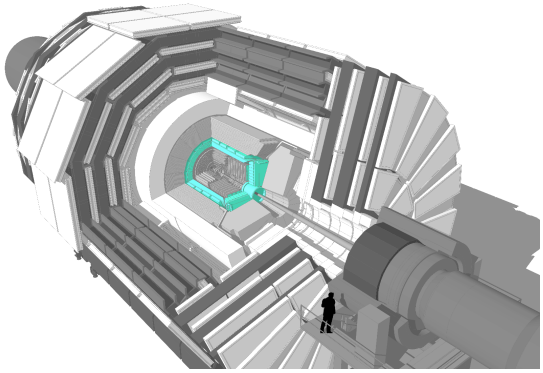
Configuración del detector

La configuración del CMS se explica a continuación [19].

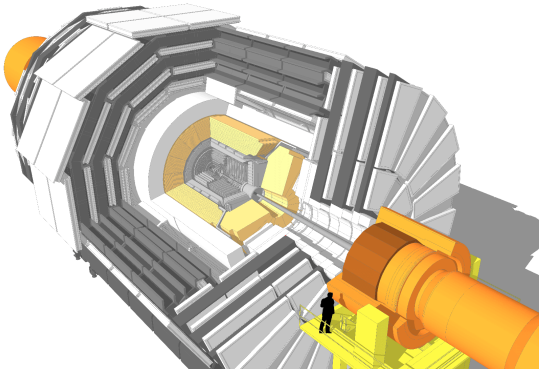
En el corazón del CMS se sitúa un superconductor solenoidal de 13 m de largo y



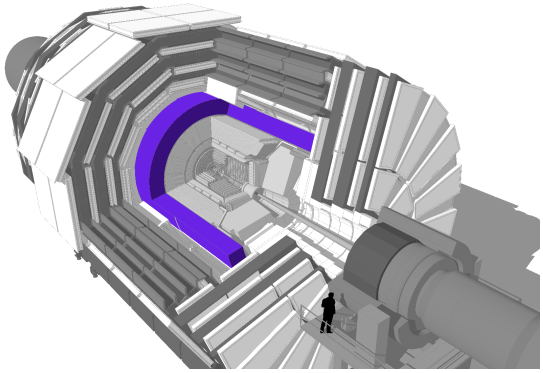
(a) *Tracker*



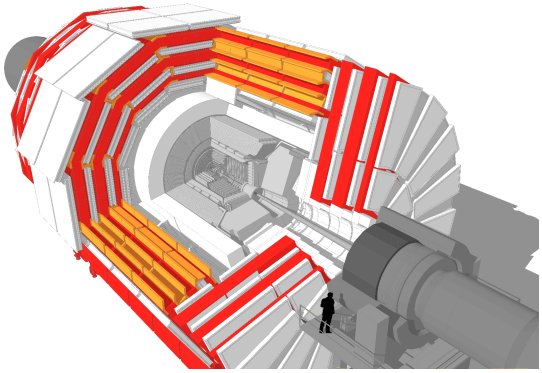
(b) Calorímetro Electromagnético



(c) Calorímetros Hadrónicos



(d) Solenoide Superconductor



(e) Sistema Muónico

Figura 2.4: Distribución de los sub-detectores dentro de CMS [3].

5.9 m de diámetro interno (figura 2.4d), el cual produce un campo magnético de 3.8 Teslas. Dentro del solenoide se encuentra el *tracker* y los calorímetros.

El Calorímetro Electromagnético (ECAL) es un calorímetro hermético y homogéneo, compuesto por cristales de plomo-tungsteno ($PbWO_4$) y cuenta con una cobertura en la pseudorapidez de hasta $|\eta| < 3.0$ (figura 2.4b). Este calorímetro consta

de dos partes, la primera es el *Electromagnetic Barrel* (EB) el cual tiene una forma cilíndrica con un radio interno de 129 cm y cubre un intervalo de pseudorapidez de $0 < |\eta| < 1.479$. La segunda parte son los *Electromagnetic Endcap* (EE) los cuales tienen forma de disco, se sitúan a una distancia de 314 cm del vértice, en dirección z, y cubren un rango de pseudorapidez de $1.479 < |\eta| < 3.0$.

El Calorímetro Hadrónico (HCAL), se localiza entre el ECAL y el solenoide magnético. Este calorímetro está compuesto por 4 partes (figura 2.4c). La primera de ellas es el *Hadron Barrel* (en color anaranjado claro en la figura 2.4c) el cual cubre una región de pseudorapidez de $0 < |\eta| < 1.4$ y tiene una segmentación de $\Delta\eta \times \Delta\phi = 0.087 \times 0.087$. Por otro lado tenemos los *Hadron Endcap* (HE) que se encuentran en cada uno de los extremos del HB y cubren una región de $1.3 < |\eta| < 3.0$ (en color anaranjado claro en la figura 2.4c). Después, tenemos el *Hadron Outer* (HO) el cual se encuentra dentro del sistema muónico y por lo tanto constreñido por la geometría y construcción de dicho sistema. El HO cubre una región de $0 < |\eta| < 1.26$ y ayuda a mejorar la resolución y la medición de la E_T^{miss} en el HCAL (se muestra color anaranjado en la figura 2.4e).

El sistema detector de muones en el CMS (de color rojo en la figura 2.4e) tiene 3 funciones principales: *triggering*, identificación y medición de momento. Este sistema cuenta con una región de *barrel* y dos *endcaps*, y se encuentran fuera del solenoide magnético. El *barrel* cubre un región de $0 < |\eta| < 1.2$ y el sistema muónico completo cubre hasta $|\eta| < 2.4$ [21].

Pese CMS no es perfecto, por ejemplo, una parte del momento llevado por los hadrones neutros es reconstruida como si fuera proveniente de fotones debido a los depósitos de energía producidos por los hadrones en el ECAL. En particular, los hadrones (neutros o cargados) depositan aproximadamente el 15 % de su energía en el ECAL [22].

2.2.4. Eventos en CMS

Un "evento" en el registro de datos del CMS se refiere a la información reconstruida y procesada acerca de los productos de una colisión de partículas (en esta tesis colisión protón-protón). Estos eventos son reconstruidos usando el algoritmo *Particle Flow* [22], el cual reconstruye e identifica partículas individuales mediante una combinación optimizada de toda la información de los sub-detectores [23].

2.2.5. Convención de unidades para *software*

El código de software de CMS utiliza las siguientes convenciones [24]:

La Energía es medida en unidades de GeV, de esta manera el momento y la masa son dadas en $\frac{GeV}{c}$ y $\frac{GeV}{c^2}$, respectivamente. La distancia y posiciones están en centímetros (cm), mientras que el tiempo está denotado en nanosegundos (ns). El campo magnético es medido en Teslas (T) y su orientación coordenada es tal que el campo magnético solenoidal está a lo largo del eje z. La carga eléctrica es medida en unidades de carga elemental $|e|$.

2.2.6. Portal de Datos Abiertos del CERN

El 20 de Noviembre de 2014 el CERN lanza su Portal de Datos Abiertos (*Open Data Portal*) donde datos provenientes de colisiones producidas en los experimentos del LHC fueron puestos accesibles al público en general por primera vez en la historia [25].

Este portal difunde materiales de diversas investigaciones, incluido software y documentación complementaria que son necesarias para entender y analizar los datos que están siendo compartidos [26]. Actualmente cuenta con más de 2 petabytes de datos abiertos de física de partículas.

Este portal es accesible mediante cualquier navegador web a través de la página web:

`http://opendata.cern.ch/`

En esta página podemos seleccionar el tipo de información que se desea explorar, así como centrarse en algún experimento en particular, como ALICE, ATLAS, CMS o LHCb.

2.2.7. Datos abiertos de CMS

Al ser CMS un detector en el CERN, se ha puesto para el acceso libre distintos formatos de datos que contienen diversos grados de detalles, tamaño y refinamiento para su uso en múltiples escenarios. A su vez, estos datos se agrupan dentro de archivos con múltiples formatos de eventos, de acuerdo con el origen o contenido de estos.

En el portal de datos abiertos del CMS podemos encontrar conjuntos de datos, documentación, software, herramientas de visualización de eventos, entre otros.

Estructura y formato de los datos

Los datos de CMS se organizan en una jerarquía de niveles de datos. Cada evento de física se escribe en cada nivel de datos, donde los niveles contienen un tipo diferente de información sobre el evento. Los tres niveles principales para datos de CMS son [27]:

1. RAW: Contiene información *cruda* proveniente del detector (señales en los elementos del detector). Este tipo de datos no es utilizado para hacer análisis.

2. RECO: Obtiene su nombre de *reconstructed data*, es el primer paso de procesamiento de datos. Este nivel contiene objetos de física (productos de decaimiento como muones, fotones, electrones, hadrones, jets, etc.) reconstruidos pero la información que provee sigue siendo muy detallada. Este tipo de datos puede ser utilizado para análisis pero es demasiado grande, o pesado, para su uso frecuente cuando CMS ha almacenado una muestra de datos sustancial.

3. AOD: El *Analysis Object Data* es la versión "destilada" de RECO y se espera sea usada para análisis. AOD provee un equilibrio entre tamaño de evento y complejidad de la información disponible, ésto con el fin de optimizar la flexibilidad

y velocidad de los análisis.

La estructura de estos archivos se basa en árboles, ramas y hojas [28]. Los árboles están optimizados para reducir espacio en el disco y mejorar la velocidad de acceso. Los árboles son capaces de manejar todo tipo de datos (variables), como objetos o arreglos. La organización en ramas permite la optimización para el uso posterior, de este modo, si hay dos variables relevantes para un análisis, la manera más eficiente para ser leídas es ponerlas juntas en una misma rama. Finalmente, las variables dentro de una rama son llamadas hojas.

Conjuntos de datos utilizados

Los eventos adquiridos por el CMS son organizados en conjuntos de datos primarios. En la página de datos abiertos del CERN[5] [26] podemos encontrar diferentes tipos de AODs los cuales se especializan en distintos tipos de información.

Durante el desarrollo de este trabajo únicamente se encontraban disponibles datos de los años 2010, 2011 y 2012. En este trabajo se considerarán los AODs *Jet* [29]. Este conjunto de datos hace referencia a eventos en donde se han generado uno o más jets debido a la hadronización de cualquier partón. Los AODs *Jet* se enfocan específicamente en jets y su comportamiento, y únicamente se encuentran disponibles en los conjuntos de datos correspondientes a 2010 y 2011. Debido a que la cantidad de *data* disponible para 2010 resulta ser menor a la de 2011, se decidió utilizar datos correspondientes a 2011, que corresponden a colisiones llevadas a cabo con una energía en el centro de masa de 7 GeV.

En total se descargaron 800 AODs del conjunto `Jet`, ocupando aproximadamente 3.5 TB en almacenamiento.

2.3. Jets

Los jets son los chorros colimados de partículas que resultan de la fragmentación de un *quark*, o gluón, energético [30] en colisiones de partículas de alta energía. En este

[5] `opendata.cern.ch`

sentido, la reconstrucción de jets juega un papel muy importante puesto que permite obtener, de manera indirecta, información sobre los constituyentes más fundamentales de la materia hadrónica: los *quarks* y los gluones[31].

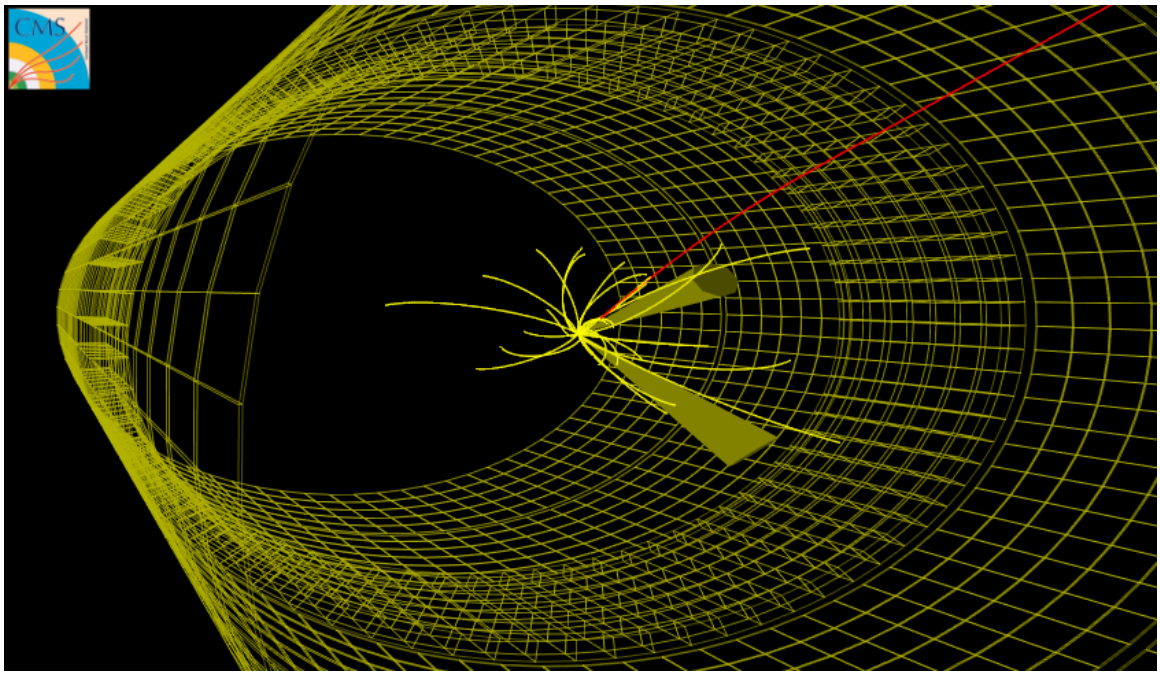


Figura 2.5: Visualización de un eventoen CMS utilizando herramientas del portal de datos abiertos del CERN [4]. En la figura se muestran las trayectorias de las partículas cargadas (líneas amarillas), un muón (línea roja) y dos jets (conos amarillos).

Para cualquier herramienta que sea utilizada para la reconstrucción de jets, su comportamiento debe de ser bien definido y reproducible: uno debería tener reglas que proyecten un conjunto de partículas dentro de un conjunto de jets. Dicho conjunto de reglas es referido como un **algoritmo de reconstrucción de jets**. Usualmente, un algoritmo de jets involucra uno o más parámetros que gobiernan su comportamiento detallado y la combinación de un algoritmo de jet y sus parámetros es conocida como una **definición de jet** [30].

El contenido de partículas de los jets, en términos de tipo de partícula y distribución de energía, puede ser descrito mediante las funciones de fragmentación y depende del sabor del partón que inicializa el jet. En promedio, 65 % de la energía del jet es llevada por partículas con carga eléctrica, 25 % por fotones y 10 % por hadrones neutros [22]. Y en promedio, los jets de gluones presentan más partículas de baja energía

que los jets de *quarks* [32].

2.3.1. Algoritmos de reconstrucción de jets

Se introducen las distancias d_{ij} y d_{iB}. Donde d_{ij} representa la distancia entre las entradas (partículas - `pseudojets`[6]-) i y j; d_{iB} es la distancia en el espacio de momentos entre la entrada i y el haz[7] B [33] y [34].

$$d_{ij} = \text{mín}\left[k_{ti}^{2p}, k_{tj}^{2p}\right]\left(\frac{\Delta_{ij}}{R^2}\right)^2 \tag{2.1}$$

$$d_{iB} = k_{ti}^{2p} \tag{2.2}$$

en estas ecuaciones k_{tn} (con $n \in i, j$) se refiere al momento transverso de la entrada n, $\Delta_{ij} = \sqrt{(\eta_i - \eta_j)^2 + (\phi_i - \phi_j)}$ corresponde a la distancia en el espacio $\eta - \phi$ entre las entradas i y j, y R es el radio del jet, que es un parámetro escalar. El parámetro p gobierna la potencia relativa de la energía contra las escalas geométricas (Δ_{ij}).

El algoritmo de reconstrucción procede a identificar la menor de las distancias, si está es d_{ij} entonces las entradas i y j son recombinadas, por el otro lado si es d_{iB} entonces llamamos a i como un jet y lo quitamos de la lista de entradas. Finalmente, estas distancias son recalculadas y el procedimiento repetido hasta que todas las partículas sean parte de algún jet y la distancia entre los ejes de todos los jet sea mayor a R (reconstrucción inclusiva). O hasta que un número deseado de jets ha sido encontrado (reconstrucción exclusiva).

Algoritmo *kt*

Considerando la ecuación 2.1 con el parámetro $p = 1$ obtenemos el algoritmo kt.

$$d_{ij} = \text{mín}\left[k_{ti}^{2}, k_{tj}^{2}\right]\left(\frac{\Delta_{ij}}{R^2}\right)^2 \tag{2.3}$$

[6]En términos computacionales, un `pseudojet` es un conjunto de partículas que son consideradas para formar jets.

[7]En este caso nos referimos al haz de partículas colisionantes.

Sin embargo, en términos generales para cualquier $p > 0$ obtendremos el mismo comportamiento porque lo que importa es el ordenamiento entre las entradas, y para todo Δ_{ij} finito, este ordenamiento es mantenido para todo valor positivo de p.

Como podemos ver en la ecuación (2.3), en el parámetro mín $\left[k_{ti}^2, k_{tj}^2\right]$ tenemos preferencia para momentos transversos bajos. Por lo tanto, el algoritmo kt prefiere agrupar primero las partículas de bajo momento transverso en una región que fluctúa considerablemente al inicio.

2.3.2. Tipos de jets

Cuando se clasifican los jets se deben de tomar en cuenta lo siguiente [35]:

- **El algoritmo de reconstrucción de jets**: Estos algoritmos hacen referencia a los criterios utilizados para asociar las partículas a los jets. Dos ejemplos de estos algoritmos son *kt* y *ak* [33], [34].
- **El radio del jet**. Es decir, el parámetro de distancia utilizado para agrupar las partículas dentro de un mismo jet, este parámetro está medido en términos de las coordenadas angulares ϕ y η. De tal modo que $R = \sqrt{\Delta\phi + \Delta\eta}$.
- **El tipo de partículas utilizadas**. Por ejemplo, PF es un algoritmo utilizado para reconstruir las partículas en un detector pero no es el único, otro algoritmo es el *Calo*, el cual utiliza únicamente información proveniente de los calorímetros. Otro ejemplo son las partículas generadas mediante simulaciones.

De este modo, `kt6PFJet` es un jet reconstruido con el algoritmo *kt* cuyo parámetro de distancia de $R = 0.6$, y que utiliza partículas reconstruidas con el algoritmo PF.

2.4. Eventos DiJet y MultiJet

El objetivo final de este trabajo es ser capaces de realizar una clasificación de eventos en donde se encuentren presentes jets. En específico, los eventos de interés son aquellos en donde la energía y la disposición espacial de los jets es tal que se pueden categorizar en dos clases específicas: DiJet y MultiJet.

2.4.1. DiJet

Los eventos DiJet se caracterizan por estar compuestos de al menos dos jets. Al jet de mayor energía se le conoce como principal o líder, mientras que al segundo más energético se le conoce como subprincipal o sublíder. Las características cinemáticas para que un evento sea considerado como DiJet son las siguientes [23], [36], [37]:

- El momento transverso p_T de los dos jets principales (1st y 2nd) debe de ser mayor a 50GeV.
- $p_T^{\text{1st}}/p_T^{\text{2nd}} < 1.5$.
- Al menos uno de los jets principales debe de estar en $|\eta| < 1.3$.
- Separación azimutal ente los dos jets principales $\Delta\phi(\text{1st}, \text{2nd}) > 2.7$.
- En caso de existir un 3er jet, este debe tener un $p_T^{3rd} < 5.0GeV$ y cumplir la condición:

$$\frac{2p_T^{3rd}}{p_T^{\text{1st}} + p_T^{\text{2nd}}} < 0.2$$

2.4.2. MultiJet

Los eventos MultiJet se caracterizan por tener un jet principal cuyo $p_T > 250GeV$ y $|\eta| < 1.3$, balanceado por un sistema de retroceso compuesto por dos o más jets cuyos $25 < p_T < 750GeV$ y satisfacer la condición $p_T^{\text{2nd}}/p_T^{\text{retroceso}} < 0.6$. Además, la distancia en el ángulo azimutal entre el jet principal y cualquiera de los jets de retroceso debe ser $|\Delta\phi(\text{1st}, \text{retroceso})| < 2.8$ [23].

2.5. Root

Root es un marco de trabajo orientado a objetos y dirigido a resolver los retos de análisis de datos en física de altas energías [28]. Proporciona las funcionalidades necesarias para tratar el procesamiento de *big data*, y el análisis estadístico y la visualización de datos [38].

2.5.1. Archivos `.root`

En general, la organización de las bases de datos consiste en un modelo en donde se tienen copias de la misma estructura de datos (generalmente llamado "*record*"), dando como resultado un arreglo bidimensional (generalmente llamada una "tabla"). En el caso de Root, estas tablas son llamadas "n-tuplas" y a los *records* se les llama "eventos"; mientras que a las columnas son llamadas variables [28].

En términos de codificación de datos, Root utiliza el formato binario para el almacenamiento. Esto permite un uso más eficiente del espacio de almacenamiento.

En el caso de la estructura de almacenamiento, su característica principal es que evita almacenar por separado la información de cada evento. En su lugar almacena la información correspondiente a la misma variable, de los distintos eventos, en la misma columna.

Por otra parte, Root permite organizar la información en estructuras similares a la estructura de directorios de cualquier sistema operativo. Siendo el directorio principal el "árbol" y los distintos subdirectorios las "ramas", terminando con las variables que se encuentras almacenadas en las "hojas".

La información contenida en los archivos `.root` puede ser entendida como se muestra en la figura 2.6.

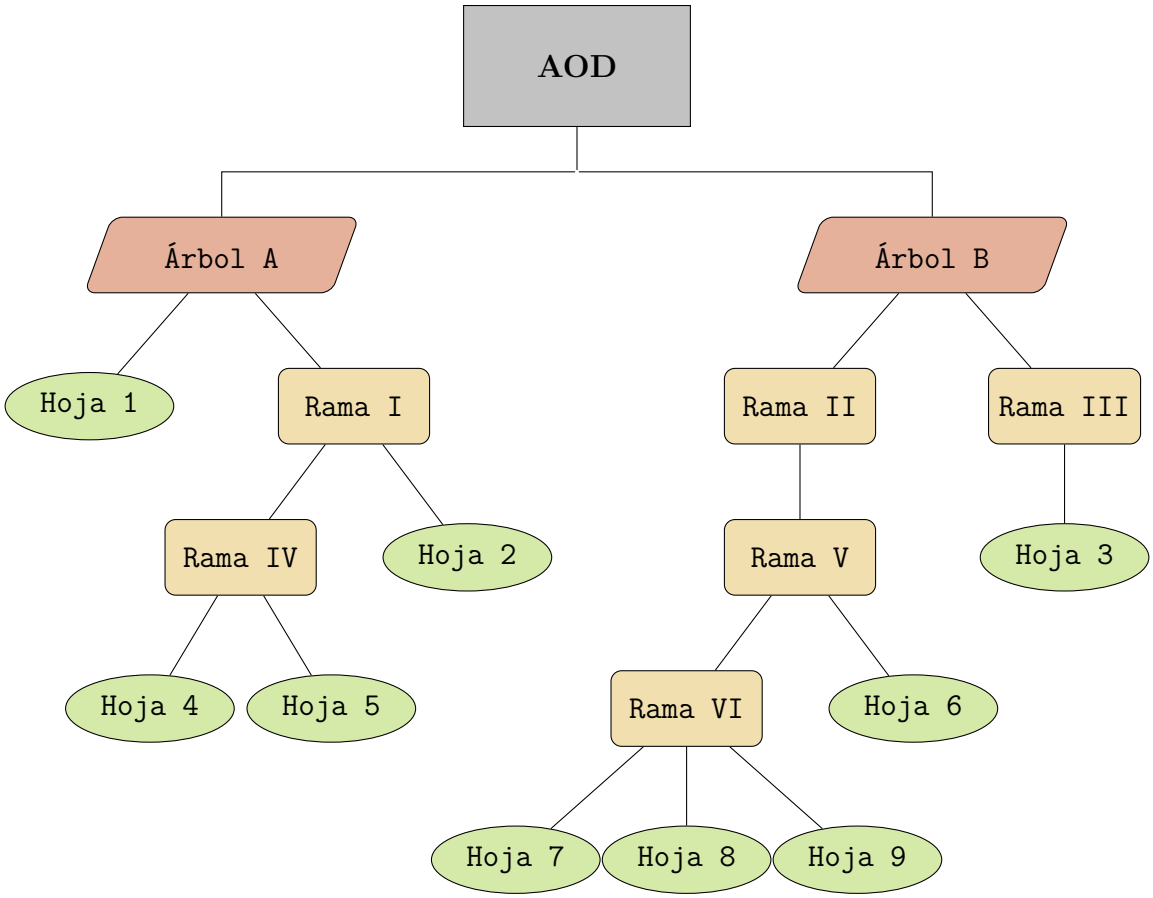


Figura 2.6: Representación de la disposición de los datos en un archivo `.root`.

Capítulo 3

Estado del Arte

En este trabajo, el *Arte* hace referencia a la clasificación de eventos. En este sentido, existen dos corrientes a considerar. La primera hace referencia a la forma metodológica tradicional, que se basa en una reconstrucción cuidadosa y bien detallada las partículas que constituyen a cada evento. Ciertamente, ésta metodología ha sido muy exitosa, pues ha conducido a logros tales como el descubrimiento del bosón de Higgs. Sin embargo, seguir éste paradigma no ha producido los evidencia esperada para inferir la existencia de nueva física. Y dado que la existencia de esta nueva física nos podría revelar mucho a cerca del universo, entonces surge la duda sobre la metodología aplicada actualmente. Diversas propuestas se han realizado para atacar este inconveniente, entre ellas, el uso de técnicas de Inteligencia Artificial.

El camino que se ha decidido seguir en este trabajo es la metodología conocida como *End-to End*, la cual se basa en la utilización de las señales crudas de los detectores de partículas para entrenar una red neuronal artificial, la cual se encargará de realizar la clasificación de eventos.

3.1. *Particle Flow*

En términos generales, los clasificadores tradicionales de eventos y los algoritmos de reconstrucción de jets están basados primordialmente en la información obtenida a partir de partículas reconstruidas [39]. En el caso concreto del experimento CMS, estas

partículas son reconstruidas mediante un algoritmo bien definido llamado *Particle-Flow* (PF), el cual combina la información proveniente de los distintos subdetectores y la procesa para reconstruir el 4-momento de todas las partículas que se originan a partir de una colisión protón-protón. El algoritmo computacional PF consiste en los siguientes pasos [40]:

1. Obtener la información necesaria:
    - Los *clusters* de los calorímetros.
    - Los *tracks*, incluido el del sistema muónico.
    - La pre-identificación de electrones.
2. Asociación de elementos topológicamente conectados (señales, de distintos subdetectores, que están relacionadas).
3. Identificación y reconstrucción de partículas.

En esencia, se limita a una prueba de hipótesis: Uno especifica las hipótesis de señal por adelantado y evalúa la presencia de dicha señal para obtener los tipos de eventos esperados. Es decir, se utiliza una estrategia supervisada [41].

3.2. Metodología *End-to-End*

La metodología *End-to-End*, introducida por M. Andrews et al. en el año 2019, consiste en construir una red neuronal para discriminar, ya sean jets o eventos, utilizando directamente las entradas de bajo nivel[1] del detector [42]. Los artículos más relevantes se enlistan en la tabla 3.1.

En todos estos artículos, se trabaja con datos abiertos simulados del detector CMS, correspondiente al año 2012 [26]. Se utiliza una técnica basada en imágenes en donde se reconstruye la información detectada por los calorímetros (ECal y HCal) más la información reconstruida del *tracker* (pues, no existen datos de las señales de

[1]Entiéndase "bajo nivel" como con poco o nulo procesamiento, es decir, las señales crudas.

Titulo	**Editorial**	**Año**
End-to-End particle and event identification at the Large Hadron Collider with CMS Open Data [42]	ArXiv	2019
End-to-End Jet Classification of Quarks and Gluons with the CMS Open Data [39]	Elsevier BV	2020
End-to-End Physics Event Classification with CMS Open Data [43]	Springer Science and Business Media	2020
End-to-End Jet Classification of Boosted Top Quarks with the CMS Open Data [44]	American Physical Society	2021

Tabla 3.1: Estado del Arte de la metodología End-toEnd desarrollada por M. Andrews et al.

estos). Los distintos artículos tienen distintas propuestas respecto a las redes neuronales utilizadas. Sin embargo, la utilización de la red `ResNet-34` (a.k.a. `ResNet-15`, debido al año de su implementación) es omnipresente y representa la mejor opción en términos globales, con la cual se obtienen desempeños de aproximadamente 80 % de efectividad.

El primero de estos realiza clasificaciones físicas utilizando información de partículas electromagnéticas, es decir, se enfoca en discriminar entre señales generadas por un electrón de las generadas por un fotón. Únicamente se utiliza la información proveniente de los calorímetros. En el segundo, se aplica esta técnica al problema de la clasificación de la subestructura de los jets. En particular, se estudia la clasificación de jets producidos por *quarks* contra los producidos por gluones. El tercer trabajo aborda el problema de clasificar eventos en donde fotones de alta energía están presentes, los cuales provienen de distintos procesos físicos. El último trabajo se centra en discriminar jets provenientes de la hadronización del *quark top*, de aquellos que provienen de la hadronización de *quarks* más ligeros o de gluones.

Capítulo 4

Solución del problema

4.1. Descripción general de la metodología

El primer paso de este trabajo es la descarga de los datos abiertos de eventos reales correspondientes al año 2011 del experimento CMS. Posteriormente, la metodología consiste en transformar los datos numéricos contenidos en los archivos `.root`, en imágenes de tamaño 34×72 píxeles. A continuación, se entrena una red neuronal para realizar la segmentación de los jets generados en los eventos. Finalmente, se entrena una segunda red neuronal, utilizando una mezcla de datos segmentados y no segmentados, la cual clasificará los eventos.

4.1.1. Descarga de datos

Para trabajar con los datos abiertos, CMS recomienda utilizar su máquina virtual (MV) [45]. Sin embargo, durante este trabajo se prescindirá de la MV para actividades distintas a la descarga de datos (debido a que ésta tiene los permisos necesarios para acceder a, y descargar, los descargas [46], [47]). Pues las herramientas necesarias para el uso que se le dará a los datos en este trabajo pueden ser instaladas directamente en el sistema operativo. Además, al ser una máquina virtual, es incapaz de utilizar todos los recursos de la computadora en donde ha sido instalada.

Descripción de la metodología de descarga

La descripción detallada de la metodología de descarga de datos puede encontrarse en el capítulo 2 de [48]. En esta sección se da una breve descripción del procedimiento.

Lo primero que se necesita es descargar e instalar la MV de CMS para el conjunto de datos abiertos correspondientes al año 2011, este procedimiento es descrito por CERN en [45]. En este trabajo se utilizó la versión *CMS-Open-Data-1.3.0*, esta MV está basada en Scientific Linux 6. Posterior a eso debemos seguir los siguientes pasos:

Dentro de la MV, es necesario activar el entorno de *software* de CMS. Este entorno cuenta con los permisos necesarios para utilizar el protocolo `XRootD` con el cual es posible descargar los datos abiertos de CMS, mediante el comando `xrdcp`. La única información necesaria es la dirección de descarga los archivos de interés, la cual puede consultarse directamente de la página de datos abiertos, [26].

Una muestra de la descarga realizada mediante la terminal de la máquina virtual es la que se muestra a continuación:

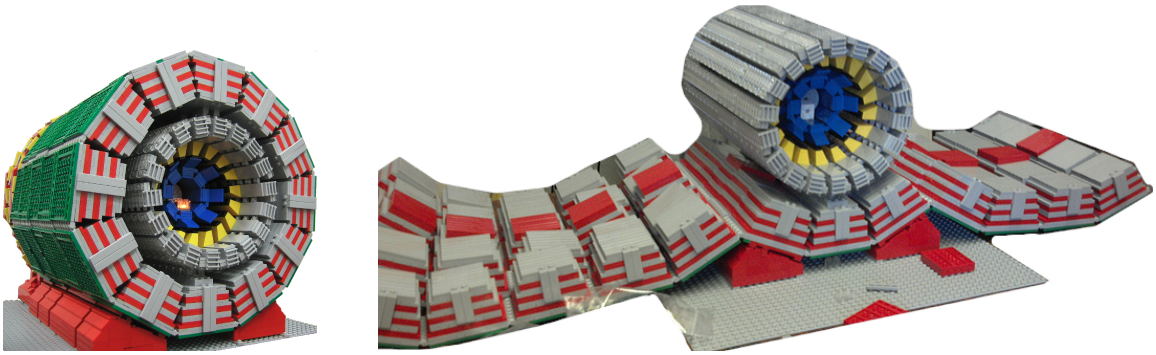
```
1 !/bin/bash
2
3 xrdcp -v -f
   ↪ root://eospublic.cern.ch//eos/opendata/cms/Run2011A/MinimumBias
   ↪ /AOD/12Oct2013-v1/20000/00658730-8546-E311-B40B-003048F23FE2.root
   ↪ /mnt/shared/Pawahtun/CMS_Run2011A/MinBias_20000/0001.root
4 xrdcp -v -f
   ↪ root://eospublic.cern.ch//eos/opendata/cms/Run2011A/MinimumBias
   ↪ /AOD/12Oct2013-v1/20000/006DE055-8646-E311-8C31-0025904B3072.root
   ↪ /mnt/shared/Pawahtun/CMS_Run2011A/MinBias_20000/0002.root
5 xrdcp -v -f
   ↪ root://eospublic.cern.ch//eos/opendata/cms/Run2011A/MinimumBias
   ↪ /AOD/12Oct2013-v1/20000/008C3D2F-9145-E311-828D-0025901AF548.root
   ↪ /mnt/shared/Pawahtun/CMS_Run2011A/MinBias_20000/0003.root
```

Las opciones adicionales `-v` y `-f` sirven para mostrar el porcentaje de descarga y para sobrescribir archivos ya existentes, respectivamente. Los datos se almacenan en un disco duro externo (llamado“Pawahtu”).

4.2. Análisis de la topología del detector

Tal como ha sido explicado en la sección 2.2.3, CMS está compuesto por una serie de secciones cilíndricas concéntricas. En particular, los dos subdetectores que nos interesan (los calorímetros electromagnético -ECal- y hadrónico -HCal-) cuentan con una geometría cilíndrica, en la parte conocida como "barril", las cuales comprenden los rangos de $[0, 2\pi]$ y $[0, 1.479]$ en ϕ y η, respectivamente.

Dada esta geometría, es posible hacer una transformación de la parte de barril de los calorímetros a un plano cuyas coordenadas estén descritas por la posición en el espacio de η y ϕ. En términos prácticos, lo que se realiza es un desdoblamiento en la coordenada ϕ de la parte cilíndrica del detector. Este desdoblamiento puede entenderse de manera simbólica como se muestra en la imagen 4.1.



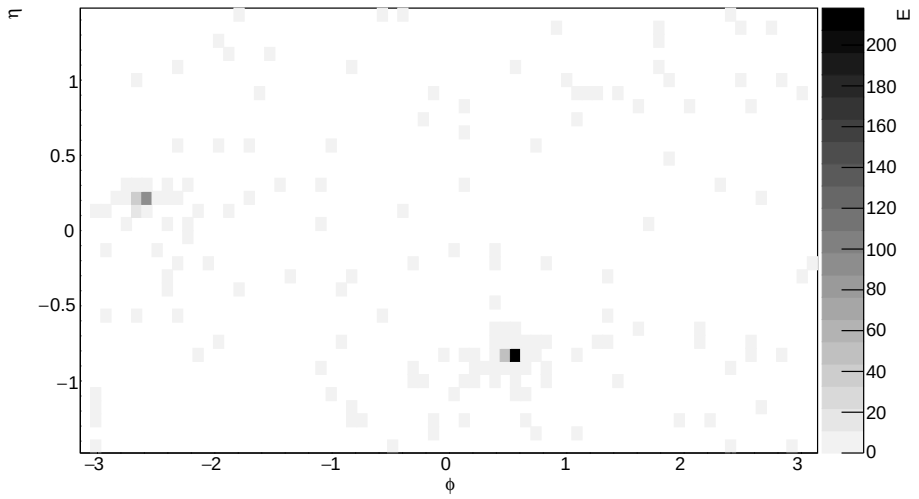
(a) CMS (b) CMS "desdoblado"

Figura 4.1: Ejemplificación figurativa del desdoblamiento en ϕ de los subdetectores del detector CMS.

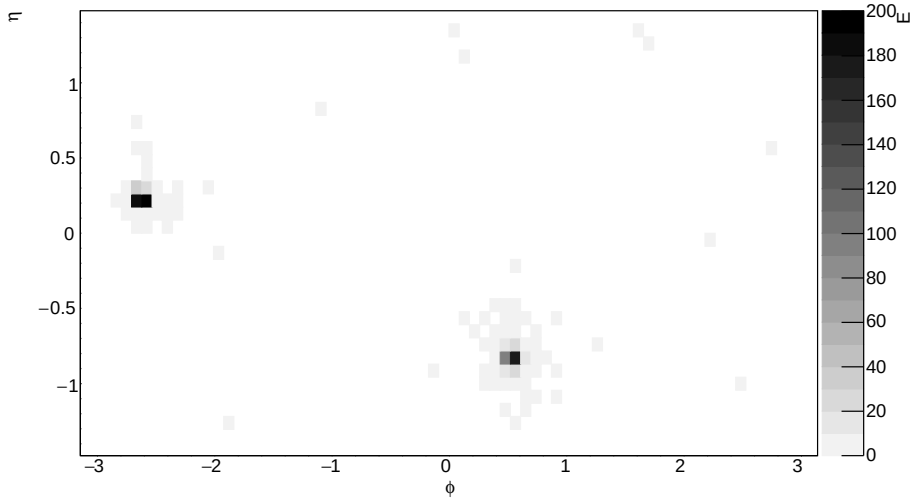
4.3. Transformación de señales a imágenes

Como se mencionó en la sección 2.5.1, Root almacena la información de cada evento en arreglos dinámicos en formato binario, dentro de los archivos `.root`. Además, debido a la naturaleza estocástica de los eventos, y por ende de las detecciones en los calorímetros, los archivo `.root` únicamente almacenan la información de las partes del detector que registran alguna señal. Consecuentemente, es necesario ex-

traer la información correspondiente a cada evento, llenar la información faltante, y transformarla a imágenes.



(a) Calorímetro Electromagnético



(b) Calorímetro Hadrónico

Figura 4.2: Distribución de la energía (E) depositada en los calorímetros.

Dado que los calorímetros (ECal y HCal) presentan una discretización en el plano $\phi - \eta$, entonces la información correspondiente a la energía depositada en ellos está discretizada de manera natural en las coordenadas ϕ y η. Sin embargo, el tamaño de estas discretizaciones varía dependiendo del calorímetro considerado. A saber, con la

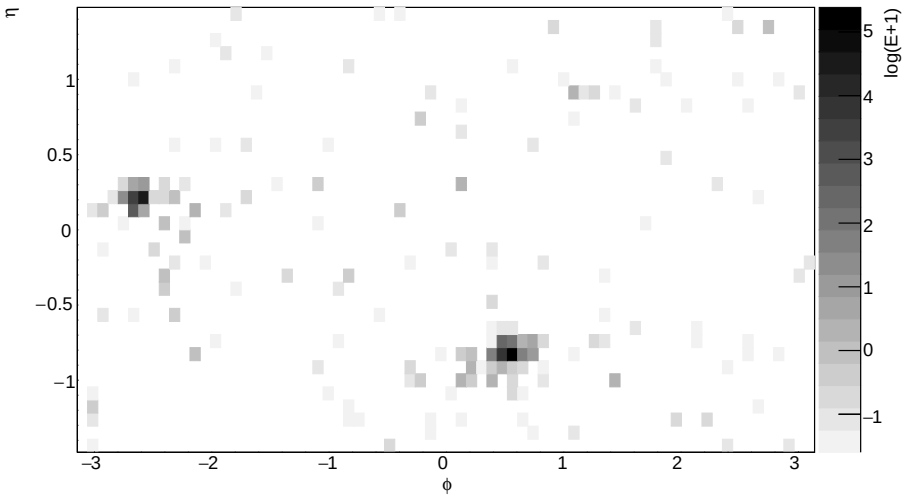
parte de barril del ECal es posible generar imágenes de tamaño 170x360 píxeles (en otras palabras, la resolución angular es $\Delta\phi \times \Delta\eta = 0.087 \times 0.087$). Mientras que en el caso del HCal, las imágenes tienen un tamaño de 34x72 píxeles (o una resolución de $\Delta\phi \times \Delta\eta = 0.174 \times 0.174$). En ambos casos, la distribución espacial cubierta es la misma.

Por tiempos computacionales, se optó por reducir el tamaño de las imágenes del ECal para que sean del mismo tamaño que las del HCal, conservando los depósitos totales de energía. Esto permitirá una menor complejidad computacional en los análisis posteriores. De este modo, una representación de la energía depositada en los calorímetros es la mostrada en la figura 4.2:

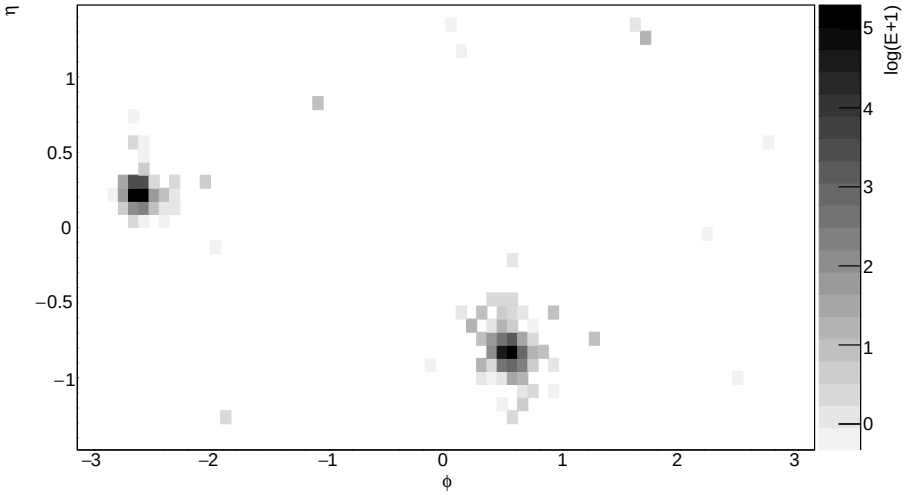
Lo primero que se nota en estos depósitos de energía es que son, relativamente, pocos los puntos que tienen la mayor concentración de energía. Esto puede resultar en un problema debido a que la intención es utilizar estas imágenes en redes neuronales, y tener unos puntos tan intensos podría sesgar el entrenamiento y funcionamiento de las redes. Por este motivo se optó por minimizar la diferencia tomando el logaritmo de la suma de la energía + 1:

$$E' = \log(E + 1)$$

Esta transformación nos lleva a las distribuciones mostradas en la figura 4.3 en donde se observan mejor los patrones generados por depósitos a baja energía.



(a) Calorímetro Electromagnético



(b) Calorímetro Hadrónico

Figura 4.3: Distribución del $\log(E+1)$ depositada en los calorímetros.

Una vez hecha esta transformación logarítmica, se procede a generar la imagen RGB correspondientes a cada evento. Esto se hace tomando las distribuciones bidimensionales del $\ln(E+1)$ correspondiente al ECal y al HCal, y normalizandolas a 255. La información del ECal corresponderá al plano rojo, mientras que la información proveniente del HCal conformará el plano verde. Finalmente, el plano azul se conformará por el promedio de los depósitos de energía del ECal y HCal, normalizada a 255.

Esto dará como resultado imágenes como la mostrada a continuación:

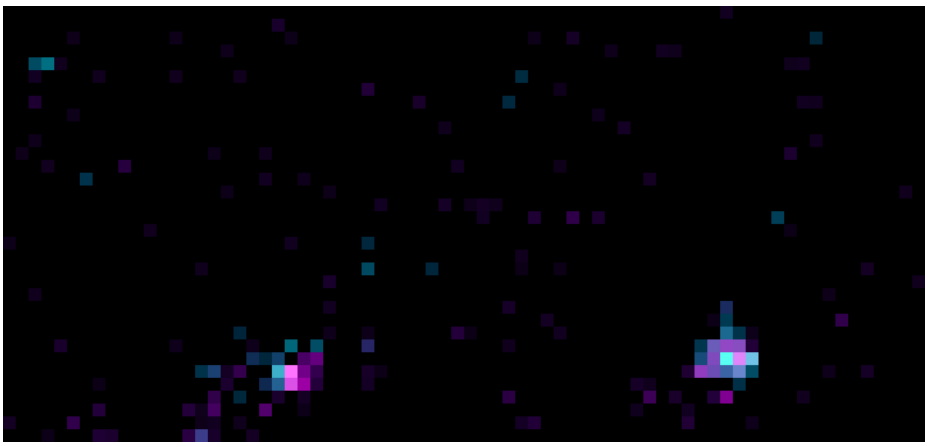


Figura 4.4: Imagen RGB correspondiente a un evento DiJet.

4.3.1. Desarrollo técnico

la generación de imágenes requiere de dos pasos: la extracción de los datos y su posterior transformación.

El entorno de trabajo Root (ver 2.5) cuenta con las paqueterías necesarias para hacer uso de los datos en formato `.root`. La extracción de datos comienza por analizar cada evento contenido en la base de datos `Jet` con el fin de discernir cuales de los eventos son DiJet, MultiJet (ver Sección 2.4) u otro tipo de eventos que en este trabajo se conocerá como Random[1]. Estas tres clases serán seleccionados partir de la información reconstruida de los eventos y se usarán para entrenar y probar el desempeño del clasificador propuesto. Por su naturaleza, estas clases son mutuamente excluyentes.

La cantidad de eventos que fueron analizados es de aproximadamente 2×10^7, repartidos en 800 archivos. Mientras que la cantidad de jets evaluados para considerar si la topología del evento cumple con las características DiJet o MultiJet es de aproximadamente 9×10^7. El pseudo-código asociado a la extracción de información de los eventos relevantes para generar la base de datos con la cual se trabajará es el

[1]La clase Random esta conformada por eventos en los cuales aparecen de 0 a 7 jets, de manera balanceada.

siguiente:

```
1  Cargar el archivo .root
2  Generar los archivos de salida .txt para cada clase
3  Cargar la información necesaria de los archivos .root
4
5  Definir variables globales
6  Definir histogramas que almacenarán la información de los eventos
   ↪ dependiendo de su clase
7
8  Para todos los eventos almacenados en el archivo .root:
9      Evaluar la cantidad y topología de los jets
10     Guardar la información del evento dependiendo de su clase en el .txt
       ↪ correspondiente
```

Donde los eventos son evaluados en función de las características cinemáticas de los jets (ver Sección 4.3.1) y a partir de estas características, se clasifican como DiJet, MultiJet o Random. Posteriormente se escriben en el archivo `.txt` correspondiente a la clase que pertenece. Es decir, se clasifican los datos y son trasformados de formato binario a UTF-8. La información que se escribe en estos archivos `.txt` corresponde a la energía medida por el ECal y el HCal en toda su constitución.

Durante el proceso de evaluación de la cantidad y topología de los jets (linea 9 del código anterior), se hace uso de la información de los jets contenida en los archivos `.root` (ver sección 4.3.1).

El tamaño total de las clases generadas se ve limitada al tamaño de la menor clase (dado que se busca que éstas estén balanceadas). En este sentido, se encontró que a la clase MultiJet pertenecen 1 de cada 650 eventos pertenecientes al conjunto `Jet`, aproximadamente.

Finalmente, para obtener las imágenes con las que se trabajarán, se utiliza un macro escrito en Python, el cual utiliza la paquetería `PIL` y convierte los datos contenidos en los archivos `.txt`, arriba mencionados, en imágenes. El procedimiento puede entenderse mediante el siguiente pseudo-código.

```
Importar las paqueterías necesarias
Definir variables globales
Cargar la información de los archivos .txt
Para cada clase:
    Para cada evento:
        Transformar los datos del ECal a escala logarítmica
        Transformar los datos del HCal a escala logarítmica
        Normalizar los datos del ECal y HCal utilizando 256 valores y un
        ↪ valor máximo de 255
        Generar una imagen RGB
        Si se utiliza para entrenar a los segmentadores:
            Generar archivo .xml
```

Si es una imagen con la cual se entrena a los segmentadores, entonces se genera un archivo `.xml` en el cual se encuentra la información referente a la ubicación de todos los jets en dicho evento (ver Sección 4.4.2).

Información de los jets

La parte medular de este trabajo consiste en la información de los jets contenida en los eventos. Esto será vital para la segmentación de los mismos y para la posterior clasificación de los eventos.

Además de las señales crudas de los calorímetros, dentro de los archivos `.root` se encuentra información de los jets reconstruidos mediante el algoritmo *kt* con radio 4 y utilizando partículas reconstruidas mediante PF (rama `reco_PFJets_kt4PFJets__RECO.obj`). La información disponible de los jets corresponde a:

- Momento transverso (p_T)
- Posición espacial (ϕ, η)
- Multiplicidad de las partículas (muones, electrones, fotones, y hadrones neutros y cargados) dentro de los jets
- Energía de las partículas (muones, electrones, fotones, y hadrones neutros y cargados) que conforman el jet

Dada la naturaleza técnica en la cual son reconstruidos los jets (ver sección 2.3.1), es necesario discernir entre jets generados por la hadronización de un partón, de aquellos que no lo son. Es decir, es necesario realizar una selección previa de los jets que se tomarán en cuanta de cada evento. Esta selección se basa completamente en restricciones cinemáticas y de constituyentes de los jets. El experimento CMS [37], [49] recomienda las siguientes restricciones:

	Holgado	**Medio**	**Estricto**
Facción de energía de hadrones neutros	<0.99	<0.95	<0.90
Fracción de energía de fotones	<0.99	<0.95	<0.90
Número total de constituyentes	>1	>1	>1
Fracción de energía de hadrones cargados	>0.00	>0.00	>0.00
Fracción de energía de electrones	>0.99	>0.99	>0.99
Número de constituyentes cargados	>0	>0	>0

Tabla 4.1: Restricciones cinemáticas para considerar que un jet proviene de la fragmentación de un partón.

En este trabajo se utilizaron las restricciones estrictas debido a que esto facilita conseguir un conjunto de clases balanceado.

4.4. Implementación de las redes neuronales

La implementación de las redes neuronales consiste principalmente en dos pasos: segmentación y clasificación.

- Primero, se proponen tres redes neuronales pre-entrenadas las cuales realizarán una segmentación de los patrones a reconocer, los jets (los cuales serán la base de la clasificación).

- Posteriormente, se construye una red neuronal que utilice la información segmentada y realice una clasificación de eventos en clases DiJet, MultiJet y Random.

4.4.1. Descripción de las redes neuronales propuestas

Las redes neuronales construidas y empleadas en este trabajo utilizan las capas residuales propuestas en [10]. Puesto que la información contenida en las imágenes es propensa al problema da degradación de patrones cuando se procesa a través de redes neuronales profundas. Esto es debido a que los patrones que se desean reconocer son de un tamaño menor a 10×10 píxeles dentro de las imágenes de tamaño 72×34 y la demás información es poco relevante.

Segmentación

Para la parte de la segmentación se utilizan 3 redes previamente entrenadas[2]: `ResNet50` [10], `ResNet101` [10], `Inception_ResNet` [50].

Los requisitos de *software* de estas redes son los siguientes:

- Python 3.6
- Tensorflow 1.2
- Numpy 1.14
- Pillow 5
- Pandas 0.25
- Matplotlib 3.5

Dada la versión específica de algunas de las paqueterías, se recomienda utilizar un ambiente dedicado de Anaconda.

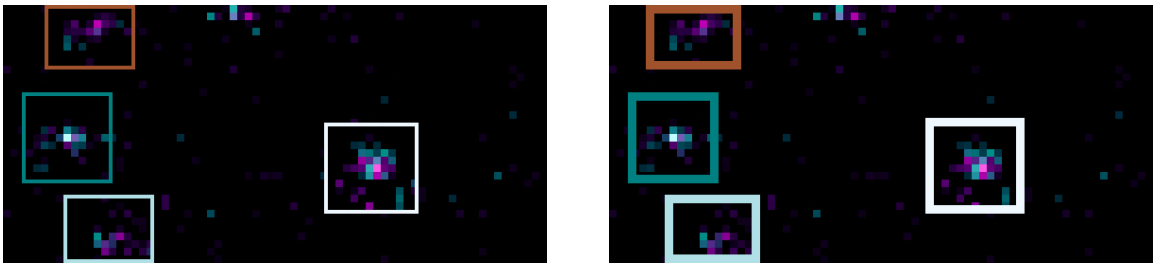
Estas tres han sido entrenadas con la base de datos COCO (Common Object Detection) de Microsoft y son capaces de detectar hasta 90 clases distintas de imágenes. Todas ellas regresan una identificación de objetos mediante un rectángulo, y muestran su clase y la probabilidad de coincidencia con la clase identificada (ver figura 4.5).

Para su adecuada implementación utilizando las imágenes de los eventos, se modificaron algunos parámetros de la red, tales como el tamaño de la capa de entrada, el tamaño de la imagen de salida, y el ancho del rectángulo segmentador.

[2]Estas redes pueden ser descargadas directamente del siguiente repositorio de GitHub `https://github.com/tensorflow/models/blob/master/research/object_detection/g3doc/tf1_detection_zoo.md`.

Figura 4.5: Segmentación de un objeto y su probabilidad de coincidencia. Predicción hecha con la `ResNet101` entrenada con la base de datos COCO.

El tamaño de la capa de entrada es proporcional al tamaño de la imágenes con las cuales se trabaja: 72×34 píxeles. El tamaño de la imagen segmentada de salida se propone como 2 veces el tamaño de la imagen de entrada, es decir, 144×68. Finalmente, el ancho del rectángulo segmentador será de 1 píxel y se eliminará el letrero con la clase y probabilidad de coincidencia. Estas dos últimas condiciones son propuestas para evitar que el rectángulo segmentador eclipse información relevante dentro de la imagen, tal como muestra la siguiente figura.



(a) 1 píxel (b) 2 píxeles

Figura 4.6: Comparación entre el ancho del rectángulo segmentador de jets. A la izquierda el ancho es de 1 píxel, a la derecha, el ancho es de 2 píxels.

Clasificación

Para el proceso de clasificación se propone una red neuronal que toma como pauta los bloques residuales propuestos en [10]. Esta elección es debido a que la topología resulta ser adecuada para la naturaleza de las imágenes con las que se trabaja: imá-

genes con patrones dispersos y relativamente "poca" información. Pues, este tipo de imágenes corre el riesgo de perder información a cada paso dentro de una red neuronal profunda. La arquitectura de los bloques utilizados se ilustra en la figura 4.7.

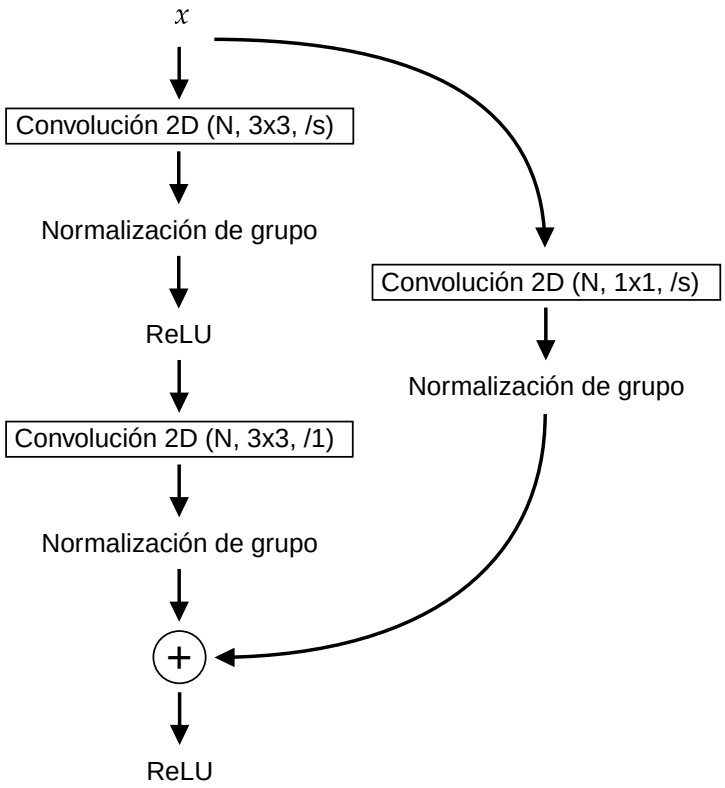


Figura 4.7: Estructura de los bloques residuales en la `ResJet`. N es la cantidad de filtros utilizados y s es el tamaño del paso en la convolución.

Para definir el tamaño de los filtros y los pasos que da cada convolución, se consideró el tamaño de las imágenes con las que se trabajan (144×68), así como el tamaño de los patrones (jets) en estas (19×19).

La red neuronal propuesta para realizar la clasificación ha sido denominada como `ResJet`, y su arquitectura se basa en un procesamiento convolucional seguido de un clasificador basado en una red perceptrón. La arquitectura se ilustra en la figura 4.8.

Evidentemente, la estructura de la red está inspirada en las redes residuales tradicionales. En este sentido, podríamos aproximar la arquitectura de la `ResJet` a una `ResNet-18`. La cantidad de parámetros con los que cuenta la `ResJet` es de 8 millones, es decir, 13 millones de parámetros menos que la `ResNet-34`.

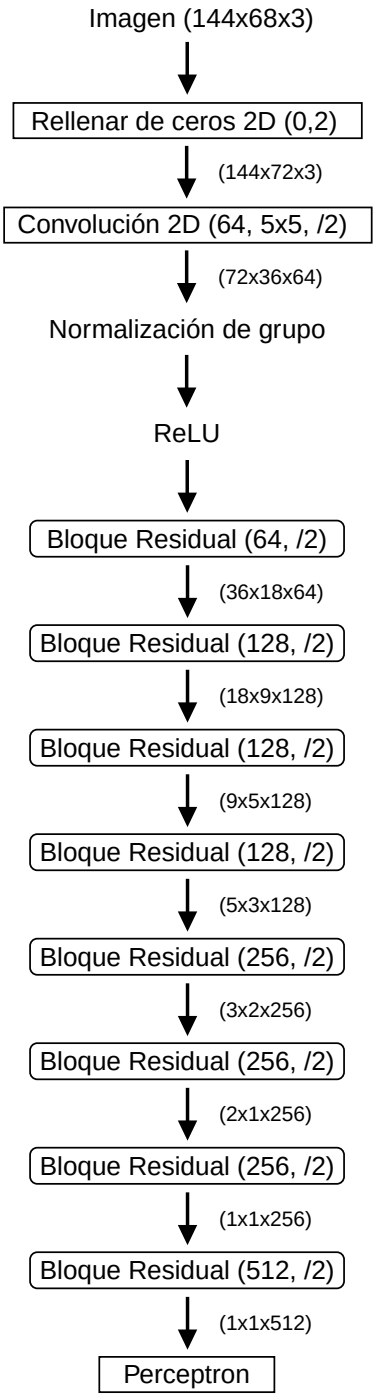


Figura 4.8: Arquitectura de la red `ResJet`.

Para construir esta red fueron tomados en cuenta los siguientes aspectos:

- El relleno inicial de ceros se realiza para que la imagen de entrada sea de tamaño 144×72. Esto es útil para el procesamiento posterior puesto que 72 tiene más

divisores que 68.

- La convolución inicial tiene como objetivo extraer las características generales de la imagen de entrada, mientras se reduce a la mitad su tamaño en ambas direcciones espaciales.

- La normalización de grupo cumple con la función de eliminar valores extremos en las características extraídas.

- Posteriormente, la función de activación $ReLU$ tiene como intención eliminar las características con valores negativos, pues éstos no son propios de las imágenes ni de lo que en particular representan en este caso (depósitos de energía).

- A continuación continúan los 8 bloques residuales. La cantidad de filtros utilizada aumenta conforme se avanza en el procesamiento. En todos los casos, el tamaño de paso de las convoluciones es de 2×2.

    - El procesamiento principal de los bloques residuales consiste en una convolución inicial de 3×3, que extrae características, mientras reduce el tamaño de los arreglos, debido al paso de la convolución que en este caso es de 2×2. Posteriormente se normalizan los valores (Normalización de grupo) y se rectifican las característica extraídas ($ReLu$).

    - Posteriormente se realiza una segunda convolución con filtros de tamaño 3×3 y paso de (1×1), y se normalizan las características extraídas

    - Por otra parte, el *skip conection* no se realiza de manera lineal, sino mediante una convolución. Esta convolución utiliza filtros de tamaño 1×1 y un paso de 2×2. Esta propuesta se utiliza debido a que la imágenes tienen patrones dispersos, y se necesita reducir el tamaño de este arreglo para que pueda sumarse con el obtenido en el punto anterior.

    - Finalmente, los resultados de estos dos procesamientos paralelos son sumados y posteriormente rectificados mediante la función de activación $ReLU$.

- Por último, el resultado obtenido de los procesos convolucionales es utilizado para alimentar a una red perceptrón mono capa, la cual se encargará de clasificar las imágenes.

4.4.2. Entrenamiento de las redes neuronales

En el caso de la segmentación, el entrenamiento de las redes se lleva a cabo mediante un conjunto de imágenes pertenecientes a las clases DiJet, MultiJet y Random. El procedimiento consiste en generar un archivo `.xml` para cada imagen en donde se especifica la ubicación y tamaño de los jets. Este archivo tiene la siguiente forma:

```
<annotation>
  <size>
    <width>72</width>
    <height>34</height>
    <depth>3</depth>
  </size>
  <segmented>0</segmented>
  <object>
    <name>Jet</name>
    <bndbox>
      <xmin>62</xmin>
      <ymin>18</ymin>
      <xmax>72</xmax>
      <ymax>29</ymax>
    </bndbox>
  </object>
</annotation>
```

En donde se especifica el tamaño y profundidad de la imagen, y los rangos en donde se encuentra(n) ubicado(s) el jet (los jets). La generación de estos archivos se realiza durante el macro que genera las imágenes RGB asociadas a cada evento (ver Sección 4.3.1).

Para entrenar a los segmentadores (`ResNet-50`, `ResNet-101` y `Inception-ResNet`), las redes deben tener acceso tanto a la imagen como al `.xml` asociados a cada evento. De este modo, el entrenamiento de las redes estará enfocado íntegramente al reconocimiento de jets.

Cada una de estas redes se entrenó durante 100 épocas y un *batch size* de 1. Los tiempos de entrenamiento variaron considerablemente debido a la arquitectura de cada una de las redes, en promedio los tiempos de entrenamiento por época fueron aproximadamente de 3, 0.9, y 0.6 horas para las redes `Inception-ResNet`, `ResNet-101` y `ResNet-50`, respectivamente.

El entrenamiento del clasificador se realizó de 3 formas distintas: utilizando únicamente datos con los jets segmentados, utilizando únicamente imágenes no segmentadas, y utilizando una mezcla de imágenes segmentadas y no segmentadas. Estas tres propuestas de conjuntos de datos se realiza para conocer cuál es la mejor proporción de datos segmentados dentro del conjunto de entrenamiento, ya que la meta última es poder realizar una clasificación de eventos sin requerir una segmentación previa.

El entrenamiento de los clasificadores se lleva a cabo durante 20 épocas, utilizando una *batch size* de 32 y el optimizador Adam. Este proceso se repite 3 veces y se considera únicamente el mejor resultado obtenido.

Datos utilizados

Los datos utilizados provienen del conjuntos de datos abiertos `Jet` (ver Sección 2.2.7). Este conjunto de datos está conformado por eventos en los cuales es altamente probable que exista cuando menos un jet generado por la hadronización de un partón. Dado que el interés se encuentra en discernir de entre todos estos eventos aquellos que sean DiJet o MultiJet (ver sección 2.4), entonces resulta necesario hacer una selección y clasificación de datos dependiendo de las propiedades cinemáticas de los jets involucrados. Con esto se obtiene un conjunto balanceado de 3 clases de datos: DiJet, MultiJet y Random. En donde Random representa eventos los cuales no son DiJet ni MultiJet y en los cuales pueden existir de 0 a 7 jets.

En total se trabaja con una base de datos balanceada de 90,000 imágenes conformada por las 3 clases mencionadas, cada una de las imágenes corresponde a un evento.

- **Segmentación**: Se tiene un total de 54,000 imágenes.

  - 18,000 son para entrenar y validar la red, 12,000 son para el entrenamiento y 6,000 son para la validación.
  - 36,000 son usadas para generar la base de datos de imágenes segmentadas que posteriormente serán usadas para entrenar al clasificador.

- **Clasificación**: Se tiene un total de 72,000 imágenes. 36,000 son las imágenes segmentadas del paso anterior y 36,000 son nuevas imágenes sin segmentar. Para el entrenamiento de la red se realizan 3 propuestas, en todas el total de datos utilizado es de 36,000 imágenes.

  - Usar 100 % de datos segmentados y 0 % de datos no segmentados.
  - Usar 50 % de datos segmentados y 50 % de datos no segmentados.
  - Usar 0 % de datos segmentados y 100 % de datos no segmentados.

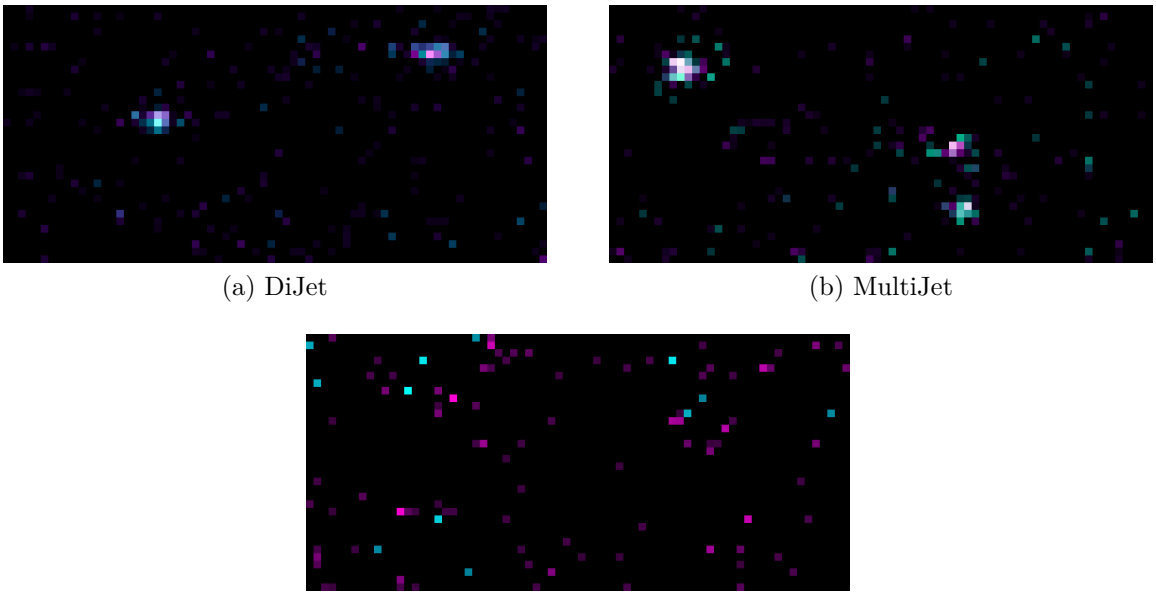
  En todos los casos, se utilizan 27,000 imágenes para entrenar a la red, y 9,000 para validar el entrenamiento.

A continuación se muestra un ejemplo de las imágenes correspondientes a cada una de las tres clases con las cuales se trabaja:

4.5. Predicciones generadas

Como se menciona en la sección anterior, la implementación de las redes segmentadoras consiste en entregarle a cada una de ellas un conjunto de 36,000 imágenes para que realice una segmentación de los jets contenidos en cada una de estas imágenes. Es decir, al final de cada evento de obtiene una copia de la imagen de entrada con los jets segmentados dentro de ella, como se muestra en la figura 4.10.

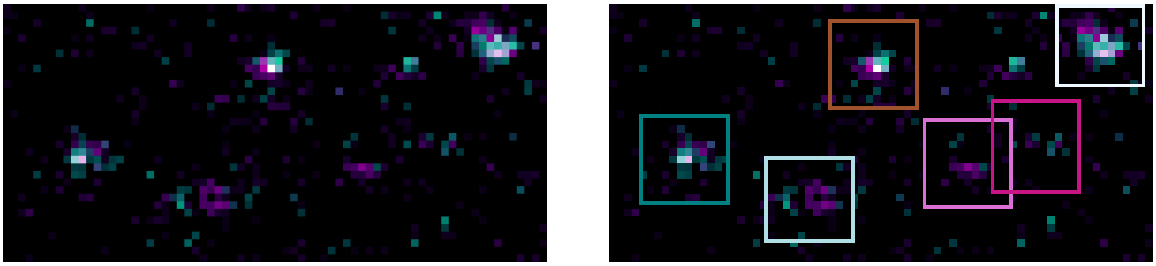
Finalmente, la red propuesta para realizar la clasificación (`ResJet`) toma como parámetro de entrada una imagen y su labor consiste en indicar la clase a la cual pertenece dicha imagen, ya sea DiJet, MultiJet o Random. Es decir, la salida de esta red es la predicción de la clase a la cual pertenece la imagen de entrada.



(a) DiJet (b) MultiJet

(c) Random

Figura 4.9: Imágenes correspondiente a cada una de las clases de eventos con las que se trabaja.



(a) Entrada (b) Salida

Figura 4.10: Imágenes correspondiente a un evento Random segmentado utilizando utilizando la red `ResNet-101`. a) Imagen de entrada a la red. b) Imagen de salida de la red con los jets segmentados.

4.6. Validación de los resultados obtenidos

4.6.1. Segmentación

Para validar los resultados obtenidos, se utilizó un conjunto de eventos de los cuales se conoce la cantidad y ubicación de jets que contienen. Se espera que las predicciones hechas por las redes concuerden con los resultados teóricos al detectar tanto la cantidad como la ubicación de los jets en cada evento.

Para dar una medida de cuan buena es el rendimiento de las redes, se utiliza el Error Cuadrático Medio (MSE, por sus sigla en inglés):

$$MSE = \frac{1}{n}\sum_{i=1}^{n}\left(\langle x_i \rangle - x_i\right)^2$$

donde la cantidad real de jets en el evento es $\langle x_i \rangle$ y la cantidad de jets detectados es x_1.

4.6.2. Clasificación

Los resultados obtenidos mediante esta metodología serán validados mediante una comparación directa con los resultados obtenidos con la red utilizada en el estado del arte, la `ResNet-34`.

La `ResNet-34` será entrenada utilizando el mismo conjunto de datos que la `ResJet`. Como métricas de desempeño serán utilizadas el *precision* y *recall*. Además, también se comparará la profundidad de las redes y la cantidad de parámetros de cada una de ellas.

Capítulo 5

Evaluación de los resultados

La meta final de este trabajo es realizar una clasificación de eventos utilizando únicamente la información proveniente de los calorímetros. Con este fin se ha propuesto una metodología basada en una previa segmentación de los patrones a detectar. Para evaluar el desempeño de esta propuesta, se entrena 3 veces a la `ResJet` utilizando una mezcla de datos segmentados y no-segmentados en distintas proporciones. Posteriormente, se evalúa el desempeño de la red haciendo uso primero de datos segmentados y después de datos no-segmentados.

5.1. Porcentaje de efectividad

5.1.1. Segmentación

Para dar un valor cuantitativo de la efectividad que tienen las redes utilizadas durante el proceso de segmentación, se utiliza el MSE. En este sentido, el error aumenta si no segmentan a todos los jets en el evento, o si se segmentan erróneamente señales que no corresponden a jets.

Para cada red, el MSE fue calculado tomando una muestra de 100 imágenes y se cotejó manualmente la cantidad y ubicación de los jets segmentados, contra los jets reales en el evento. Los resultados obtenidos se muestran en la siguiente tabla:

	ResNet-50	**ResNet-101**	**Inception-ResNet**
MSE	0.84	0.33	0.44

Tabla 5.1: MSE del desempeño en la segmentación de las distintas redes neuronales utilizadas.

Es de notar que el mejor desempeño fue obtenido por la `ResNet-101`. Por ende, los datos segmentados en la clasificación posterior provendrán de las predicciones realizadas con esta red.

5.1.2. Clasificación

Para evaluar la efectividad de la clasificación independientemente de si se utilizan datos normales o segmentados, se evalúa el valor de las métricas de exactitud (*accuracy*), precisión (*presicion*) y la retirada (*recall*), además de la matriz de confusión. En todos los casos, se ejecutó 3 veces cada red durante 20 épocas, un *batch size* de 32 y se muestra el mejor resultado obtenido.

En primer lugar se muestran las predicciones obtenidas con la `ResNet-34` entrenada únicamente con datos no-segmentados.

Datos de prueba	**Exactitud**	**Precisión**	**Retirada**	**Mat. de Confusión**
No-segmentado	0.850	0.852	0.849	$\begin{pmatrix} 1791 & 294 & 221 \\ 138 & 1611 & 107 \\ 97 & 40 & 1701 \end{pmatrix}$

Tabla 5.2: Desempeño de la `ResNet-34` al ser entrenada con datos no-segmentados. Se muestra el mejor resultado de un total de tres entrenamientos.

Los renglones de la matriz de confusión representa la clase esperada, mientras que las columnas representan la clase predicha. El primer renglón (columna) corresponde a la clase DiJet, el segundo a la MultiJet y el tercero a Random.

A continuación se muestra el desempeño obtenido al ser entrenada únicamente con datos segmentados.

Datosde prueba	Exactitud	Precisión	Retirada	Mat. de Confusión
Segmentado	0.855	0.857	0.853	$\begin{pmatrix} 1864 & 132 & 164 \\ 93 & 1672 & 229 \\ 76 & 177 & 1593 \end{pmatrix}$

Tabla 5.3: Desempeño de la `ResNet-34` al ser entrenada con datos segmentados. Se muestra el mejor resultado de un total de tres entrenamientos.

Ahora, se muestran los resultados obtenidos con la `ResJet`. Al igual que en el caso anterior, se muestra el mejor desempeño de entrenar 3 veces distintas a la red. Del mismo modo, cada red fue entrenada durante 20 épocas, usando un *batch size* de 32.

La siguiente tabla corresponde a la evaluación de la `ResJet` después de ser entrenada con datos no-segmentados.

Datos de prueba	Exactitud	Precisión	Retirada	Mat. de Confusión
No-segmentados	0.873	0.874	0.872	$\begin{pmatrix} 1807 & 183 & 202 \\ 142 & 1733 & 130 \\ 77 & 29 & 1697 \end{pmatrix}$

Tabla 5.4: Desempeño de la `ResJet` al ser entrenada y validada utilizando datos no-segmentados. Se muestra el mejor resultado de un total de tres entrenamientos.

Por último, se muestra el desempeño de la `ResJet` al ser entrenada con datos segmentados.

Datos de prueba	Exactitud	Precisión	Retirada	Mat. de Confusión
Segmentados	0.816	0.818	0.814	$\begin{pmatrix} 1787 & 194 & 217 \\ 102 & 1544 & 291 \\ 73 & 229 & 1563 \end{pmatrix}$

Tabla 5.5: Desempeño de la `ResJet` entrenada con datos segmentados. Se muestra el mejor resultado de un total de tres entrenamientos.

5.2. Limitaciones de la metodología propuesta

La primer limitación reseñable de la metodología propuesta es que se enfoca en la clasificación de eventos que son definidos exclusivamente por la topología de los jets

contenidos. Es decir, cualquier otro evento que dependa de la dinámica de otros entes físicos, además de los jets, no está considerada en esta metodología.

Por otra parte, únicamente se utiliza información limitada del detector CMS, a saber, solo se toman las señales crudas de los calorímetros. Esto significa que no se han utilizado las señales provenientes de los demás sub-detectores: el sistema *tracker* y la cámara de muones. La razón de esto es que, dentro de los archivos `.root`, no existe información a cerca de las señales crudas de estos dos últimos sub-detectores. Por lo que no pueden ser utilizados para el análisis realizado.

Finalmente, el rango utilizado de detección de los calorímetros es limitado. Pues, únicamente se toman las señales provenientes de la parte del barril ($\eta \leq 1.479$), excluyendo las partes más extremas ($1.479 < \eta < 5.0$). Sin embargo, esto no supone un problema para la aplicación que se consideró, ya que los eventos DiJet y MultiJet están bien definidos dentro de ésta área limitada.

5.3. Evaluación global de la metodología desarrollada

La evaluación de la metodología utilizada puede entenderse en términos del desempeño de la red `ResJet`, y de la mejora que se encuentra al utilizar datos segmentados para la clasificación.

En el caso del desempeño de la `ResJet`, notamos que es mejor en comparación a los resultados obtenidos con la `ResNet-34` al utilizar datos no-segmentados. Sin embargo, al utilizar datos segmentados para la clasificación, el resultado no ha sido el esperado. Pues, el desempeño ha visto una merma.

Por otra parte, se tiene que los datos segmentados han sido útiles al realizar la clasificación con la red `ResNet-34`. Ya que, su uso ha implicado una ligera mejora en los valores de desempeño.

Capítulo 6

Conclusiones y trabajos futuros

6.1. Conclusiones

Este trabajo ha tenido como intensión aportar a la metodología *End-to-End* utilizando un enfoque complementario. Pues, hasta ahora, esta metodología se caracteriza por el uso de datos simulados a la par de implementar redes neuronales previamente establecidas. En términos de las hipótesis de trabajo, se logra destacar que el uso de datos reales ha resultado factible. Pues, la clasificación de eventos tuvo un desempeño superior al 87 % al considerar la `ResJet`. Por otra parte, la segunda hipótesis que se consideró, que el uso de datos segmentados mejoraría el desempeño de las clasificaciones, ha sido cierta únicamente al utilizar la `ResNet-34`. Mientras que al utilizar la `ResJet` el desempeño ha decrecido aproximadamente en 6 %.

El primer aspecto resaltable es el hecho de haber trabajado con una base de datos correspondiente a colisiones reales de protones. Esto marca una clara diferencia con respecto al estado del arte. Pues, los datos reales contienen ruido y señales que no pertenecen propiamente a los productos de las colisiones.

En segundo lugar, debido a que los eventos que se clasificaron dependen de la cantidad y cinemática de los jets presentes, se optó por realizar una segmentación de estos, previa a la clasificación. Se propusieron 3 redes neuronales distintas, y se decantó por utilizar los datos generados por la que mejor desempeño tuvo en la labor de segmentación.

En tercer lugar, se desarrolla la red neuronal `ResJet`, la cual está basada en la arquitectura de las redes neuronales residuales, y cuyo objetivo es mejorar el desempeño de la red utilizad en el estado del arte para la clasificación de eventos. Esta meta se consigue al utilizar únicamente datos no segmentados. Por otro lado, la red utilizado en el estado del arte mejora su desempeño al realizar la clasificación usando datos segmentados.

Finalmente, el objetivo principal de este trabajo (la clasificación de eventos en las distintas clases consideradas) tuvo un desempeño superior a 0.8 en todos los casos al considerar las métricas de *precision* y *recall*. Siendo comparable a los resultados obtenidos en el estado del arte.

6.2. Trabajos futuros

Esta tesis da lugar a diversos trabajos futuros. Por ejemplo, se puede desarrollar una arquitectura específica para realizar la segmentación de los jets en las imágenes, ya que esta tarea fue realizada siguiendo la aproximación de *transfer learning*.

Por otra parte, ni en el estado del arte, ni en este trabajo se han utilizado las señales de la cámara de muones. Pues, estas no están disponibles en la base de datos pero se propone realizar una reconstrucción de estas como trabajo a futuro (similar a lo que se realiza en el estado del arte con los *tracks*).

Por otro lado, se propone modificar la estructura de la `ResJet` para poder aprovechar la información proveniente de la segmentación.

Bibliografía

[1] CERN. *When protons collide*. http://atlas.physicsmasterclasses.org/en/zpath_protoncollisions.htm.

[2] CMS Collaboration. CMS: The hadron calorimeter technical design report, 1997.

[3] David Barney and Tai Sakuma. Sketchup images highlighting the sub-detectors, Sep 2017.

[4] Colaboración CMS. *CMS Open-Data Event Display*. http://opendata.cern.ch/visualise/events/cms.

[5] Lilian Judith Sandoval. Machine learning algorithms for data analysis and prediction, 2018.

[6] A.Moreno et al. *Aprendizaje Automático*. Politext 36. Edicions UPC, Barcelona, España, 1994.

[7] Paul Wilmott. *Machine Learning*. Panda Ghana Publishing, 2019.

[8] Benjamin Johnston, Aaron Jones, and Christopher Kruger. *Applied Unsupervised Learning with Python*. Packt Publishing Ltd, Birmingham, UK, 2019.

[9] Tianyi Liu, Shuangsang Fang, Yuehui Zhao, Peng Wang, and Jun Zhang. Implementation of training convolutional neural networks, 2015.

[10] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition, 2015.

[11] C.M. Bishop, P.N.C.C.M. Bishop, G. Hinton, and Oxford University Press. *Neural Networks for Pattern Recognition*. Advanced Texts in Econometrics. Clarendon Press, 1995.

[12] M. Bachtis. *Heavy Neutral Particle Decays to Tau Pairs: Detected with CMS in Proton Collisions at* $\sqrt{s}$ *= 7TeV*. Springer Theses. Springer International Publishing, 2013.

[13] CERN. *The history of CERN*. `https://home.cern/about/who-we-are/our-history`.

[14] CERN. *What is CERN's mission?* `https://home.cern/about/who-we-are/our-mission`.

[15] Oliver Sim Brüning, Paul Collier, P Lebrun, Stephen Myers, Ranko Ostojic, John Poole, and Paul Proudlock. *LHC Design Report*. CERN Yellow Reports: Monographs. CERN, Geneva, 2004.

[16] CERN. *What is CMS?* `http://cms.web.cern.ch/news/what-cms`.

[17] CMS Collaboration. *ECAL Technical Design Report. CERN/LHCC*, Enero 2006.

[18] Clémentine Broutin. *Electron Measurements and Search for Higgs Bosons in Multi-Lepton Channels with the CMS Experiment at LHC*. PhD thesis, Ecole polytechnique, 2011.

[19] Bayatian G L, Sergey Chatrchyan, Gevorg Hmayakyan, Albert Sirunyan, Adam W, Thomas Bergauer, Marko Dragicevic, Ero J, Friedl M, Rudolf Frühwirth, Ghete V, Glaser P, Hrubec J, M Jeitler, Krammer M, Ildefons Magrans, Mikulec I, Mitaroff W, Noebauer T, and Bekhzod Yuldashev. *CMS Physics : Technical Design Report Volume 1: Detector Performance and Software. Geneva: CERN*, Enero 2006.

[20] Cheuk-Yin Wong. *Introduction to high-energy heavy ion collisions*. World Scientific, 1994.

[21] Serguei Chatrchyan et al. *The Performance of the CMS Muon Detector in Proton-Proton Collisions at $\sqrt{s}$ = 7 TeV at the LHC*. *JINST*, 8:P11002, 2013.

[22] A. M. Sirunyan et al. *Particle-flow reconstruction and global event description with the CMS detector*. *JINST*, 12(10):P10003, 2017.

[23] Vardan Khachatryan et al. Jet energy scale and resolution in the CMS experiment in pp collisions at 8 TeV. *JINST*, 12(02):P02014, 2017.

[24] T. Bergauer. *Design, Construction and Commissioning of the CMS Tracker at CERN and Proposed Improvements for Detectors at the Future International Linear Collider*. PhD thesis, Vienna, Tech. U., 2008.

[25] CERN. *CERN makes public first data of LHC experiment*. `https://home.cern/news/news/accelerators/cern-makes-public-first-data-lhc-experiments`.

[26] CERN. *CERN Open Data Portal*. `http://opendata.cern.ch/docs/about`.

[27] CERN. The CMS Offline WorkBook. In *The CMS Offline SW Guide*, chapter 2.2 Computing Model. CERN, Julio 2018. `https://twiki.cern.ch/twiki/bin/view/CMSPublic/WorkBookComputingModel`.

[28] CERN. *ROOT User's Guide*, Mayo 2018. `https://root.cern.ch/root/htmldoc/guides/users-guide/ROOTUsersGuideA4.pdf`.

[29] CMS collaboration (2016). *Jet primary dataset in AOD format from RunA of 2011 (/Jet/Run2011A-12Oct2013-v1/AOD)*. CERN Open Data Portal. `http://opendata.cern.ch/record/21`.

[30] Matteo Cacciari, Gavin P. Salam, and Gregory Soyez. *FastJet User Manual (for version 3.3.2)*. `http://fastjet.fr/repo/fastjet-doc-3.3.2.pdf`.

[31] Jeff Tseng and Hannah Evans. *Sequential recombination algorithm for jet clustering and background subtraction. Phys. Rev.*, D88:014044, 2013.

[32] The DELPHI Collaboration. *Measurement of the gluon fragmentation function and a comparison of the scaling violation in gluon and quark jets. Eur. Phys. J., C - Particles and Fields*, 13, Abril 2000.

[33] Matteo Cacciari, Gavin P. Salam, and Gregory Soyez. The anti-k_t jet clustering algorithm. *JHEP*, 04:063, 2008.

[34] Ryan Atkin. Review of jet reconstruction algorithms. *Journal of Physics: Conference Series*, 645, Octubre 2015.

[35] CERN. The CMS Offline WorkBook. In *The CMS Offline SW Guide*, chapter 7.2 Jet Analysis . CERN, Julio 2009. `https://twiki.cern.ch/twiki/bin/view/CMSPublic/WorkBookJetAnalysis`.

[36] The ATLAS Collaboration. Measurement of the charged-particle multiplicity inside jets from $$\sqrt{s}=8$$ s = 8 $${\mathrm{TeV}}$$ TeV pp collisions with the ATLAS detector. *The European Physical Journal C*, 76(6), jun 2016.

[37] The CMS collaboration. Determination of jet energy calibration and transverse momentum resolution in CMS. *Journal of Instrumentation*, 6(11):P11002–P11002, nov 2011.

[38] Rene Brun and Fons Rademakers. *ROOT - An Object Oriented Data Analysis Framework*. `http://root.cern.ch/`.

[39] M. Andrews, J. Alison, S. An, B. Burkle, S. Gleyzer, M. Narain, M. Paulini, B. Poczos, and E. Usai. End-to-end jet classification of quarks and gluons with the cms open data. *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, 977:164304, Oct 2020.

[40] CERN. Particle Flow. In *The CMS Offline SW Guide*. CERN, Noviembre 2018. `https://twiki.cern.ch/twiki/bin/view/CMSPublic/SWGuideParticleFlow#Jets`.

[41] Oliver Knapp, Guenther Dissertori, Olmo Cerri, Thong Q. Nguyen, Jean-Roch Vlimant, and Maurizio Pierini. Adversarially learned anomaly detection on cms open data: re-discovering the top quark, 2020.

[42] John Alison, Sitong An, Michael Andrews, Patrick Bryant, Bjorn Burkle, Sergei Gleyzer, Ulrich Heintz, Meenakshi Narain, Manfred Paulini, Barnabas Poczos, and Emanuele Usai. End-to-end particle and event identification at the large hadron collider with cms open data, 2019.

[43] M. Andrews, M. Paulini, S. Gleyzer, and B. Poczos. End-to-end physics event classification with CMS open data: Applying image-based deep learning to detector data for the direct classification of collision events at the LHC. *Computing and Software for Big Science*, 4(1), mar 2020.

[44] M. Andrews, B. Burkle, Y. Chen, D. DiCroce, S. Gleyzer, U. Heintz, M. Narain, M. Paulini, N. Pervan, Y. Shafi, W. Sun, E. Usai, and K. Yang. End-to-end jet classification of boosted top quarks with the CMS open data. *Physical Review D*, 105(5), mar 2022.

[45] Colaboración CMS. *CMS 2011 Virtual Machines: How to install.* `http://opendata.cern.ch/docs/cms-virtual-machine-2011`.

[46] CERN. *Getting Started with CMS 2011 Open Data.* `http://opendata.cern.ch/docs/cms-getting-started-2011`.

[47] Achim Geiser, Irene Dutta, Harri Hirvonsalo, and Bridget Sheeran. *Example code to produce the di-muon spectrum from a CMS 2011 or 2012 primary dataset*, 2017. `http://opendata.cern.ch/record/5001`.

[48] Saksevul Arias. Estudio de las propiedades de fragmentación de jets producidos en colisiones hadrónicas utilizando datos abiertos del experimento cms en el cern. Master's thesis, Universidad Nacional Autónoma de México, Ciudad Universitaria, CdMx, México, 2020.

[49] The CMS Collaboration. Jet Performance in pp Collisions at 7 TeV, 2010.

[50] Christian Szegedy, Sergey Ioffe, Vincent Vanhoucke, and Alex Alemi. Inception-v4, inception-resnet and the impact of residual connections on learning, 2016.