UNIVERSIDAD SAN FRANCISCO DE QUITO USFQ

Colegio de Ciencias e Ingeniería

# Aplicación de un modelo de clasificación de imágenes a los procesos Drell-Yan, $t\bar{t}$ + jets y W + jets y su uso en datos de colisiones reales

José David Ochoa Flores

Física

Trabajo de titulación presentado como requisito

para la obtención del titulo de

Licenciado en Física

14 de mayo de 2023

**UNIVERSIDAD SAN FRANCISCO DE QUITO USFQ**

**Colegio de Ciencias e Ingeniería**

## HOJA DE CALIFICACIÓN DE TRABAJO DE FIN DE CARRERA

Aplicación de un modelo de clasificación de imágenes a los procesos Drell-Yan, $t\bar{t}$ + jets y W + jets y su uso en datos de colisiones reales

**José David Ochoa Flores**

Nombre del profesor, Titulo académico: Edgar Carrera, PhD

14 de mayo de 2023

# © Derechos de Autor



| | |
|---|---|
| Nombres y apellidos: | José David Ochoa Flores |
| Código: | 00206156 |
| Cédula de Identidad: | 1718593278 |
| Lugar y fecha: | 14 de mayo de 2023 |

## ACLARACIÓN PARA LA PUBLICACIÓN

**Nota:** El presente trabajo, en su totalidad o cualquiera de sus partes, no debe ser considerado como una publicación, incluso a pesar de estar disponible sin restricciones a través de un repositorio institucional. Esta declaración se alinea con las prácticas y recomendaciones presentadas por el Committee on Publication Ethics COPE descritas por Barbour et al. (2017) Discussion document on best practice for issues around theses publishing, disponible en `http://bit.ly/COPETheses`

## UNPUBLISHED DOCUMENT

**Note:** The following capstone project is available through Universidad San Francisco de Quito USFQ institutional repository. Nonetheless, this project – in whole or in part – should not be considered a publication. This statement follows the recommendations presented by the Committee on Publication Ethics COPE described by Barbour et al. (2017) Discussion document on best practice for issues around theses publishing available on `http://bit.ly/COPETheses`

# Agradecimientos

Es importante empezar los agradecimientos por mis padres, porque simplemente sin ellos yo no habría podido llegar hasta acá. A mis hermanas por siempre estar dispuestas a escucharme sin importar lo que yo tuviese que contar. A mis amigos de carrera: Alejandro Rueda, Valeria Bedoya, Mateo Martínez y especialmente a María del Carmen Salazar, porque con ellos viví la bonita experiencia universitaria más allá de los salones de clase. A Juan Aguilar y Jorge Rodríguez, mis mejores amigos y a quienes les debo gran parte de quién soy. Finalmente quisiera dar un agradecimiento especial a Edgar Carrera, por estar siempre dispuesto a ayudarme, compartiendo su conocimiento y empujándome de a poco a lograr este proyecto que fue más allá de lo que hubiera pensado.

# Resumen

Existen diversas señales provenientes de procesos físicos que pueden ser similares ya sea debido a su topología o a que tienen un mismo decaimiento. Es importante, por lo tanto, encontrar maneras efectivas de distinguir los resultados finales de estos procesos. Para ello el presente trabajo emplea un modelo de clasificación de imágenes basado en redes neuronales convolucionales. Las imágenes usadas para entrenar el modelo fueron obtenidas haciendo uso de los datos abiertos del CERN. Se encontró que el modelo fue útil al clasificar entre eventos pertenecientes a los procesos Drell-Yan, $t\bar{t}$ + jets y W + jets. El modelo también fue capaz de discriminar la resonancia característica del proceso Drell-Yan en un conjunto de datos de colisiones reales. Estos resultados son alentadores para continuar explorando el alcance de estos algoritmos en la física de partículas.

Palabras clave: *clasificación de imágenes, colisiones, decaimiento, Drell-Yan, física de partículas, redes neuronales convolucionales, resonancia*

# Abstract

There are several signals coming from physical processes that can be similar either due to their topology or because they have the same decay. Therefore is important to find effective ways to distinguish between the final results of these processes. For this purpose, we have used an image classification model based on convolutional neural networks. The images that were used to train the model were obtain thanks to the CERN Open Data Portal. It was found that the model was capable of classifying between Drell-Yan, $t\bar{t}$ + jets and W + jets processes. The model was also capable of discriminating the characteristic resonance of the Drell-Yan process. This results are inspiring to keep exploring the potential of these algorithms in the context of particle physics.

Keywords: *image classification models, collisions, convolutional neural network, decay, Drell-Yan, particle physics, resonance*

# Índice general

# Índice de cuadros

# Índice de figuras

# Capítulo 1

# Introducción

Es ampliamente conocido que el foco principal de la investigación de física de partículas se encuentra en los laboratorios del CERN, en español conocido como la Organización Europea para la Investigación Nuclear. El complejo cuenta con el colisionador de hadrones más grande del planeta, el LHC, que tiene una longitud de 27 km y acelera haces de protones a velocidades cercanas a las de la luz. Estos haces chocan en lugares específicos y el resultado de estas colisiones es analizado por los distintos detectores que se encuentran alrededor del anillo, estos son: ATLAS, CMS, ALICE y LHCb. Nuestro enfoque estará en el segundo detector, el CMS. Gracias a su política de hacer públicos datos de colisiones y simulaciones, y sumando esto al auge del machine learning, la inteligencia artificial y sobretodo de los modelos de clasificación de imágenes, el presente trabajo busca explorar la utilidad de estos modelos en el panorama de la física de partículas.

## 1.1. CMS

El Solenoide de Muones Compacto, o por sus siglas en inglés CMS, es un detector multi-propósito del LHC. El detector está construido alrededor de un solenoide magnético gigante que puede generar un campo mágnetico de hasta 3.8 Teslas. El CMS es una de las colaboraciones científicas más grandes del planeta y tiene la participación de 241 institutos pertenecientes a 54 países. Para empezar a entender la física detrás de los experimentos que se realizan en el LHC es necesario entender el sistema de coordenadas que ha sido adoptado en el detector [5]. Éste está centrado en el punto de colisión, el eje-$y$ apunta verticalmente hacia arriba, y el eje-$x$ apunta radialmente hacia adentro. Por lo tanto, el eje-$z$ restante apunta en dirección de una cadena montañosa llamada Jura, en la región de Ginebra, Suiza. El ángulo azimutal $\phi$ es medido desde el eje-$x$ en el plano $xy$. El ángulo polar $\theta$ por su parte, es medido desde el eje-$z$. Haremos uso también de la pseudorapidez $\eta$, definida como

$$\eta = -\ln \tan\left(\frac{\theta}{2}\right). \tag{1.1}$$

Podemos observar la distribución de los detectores en la Figura 1.1.

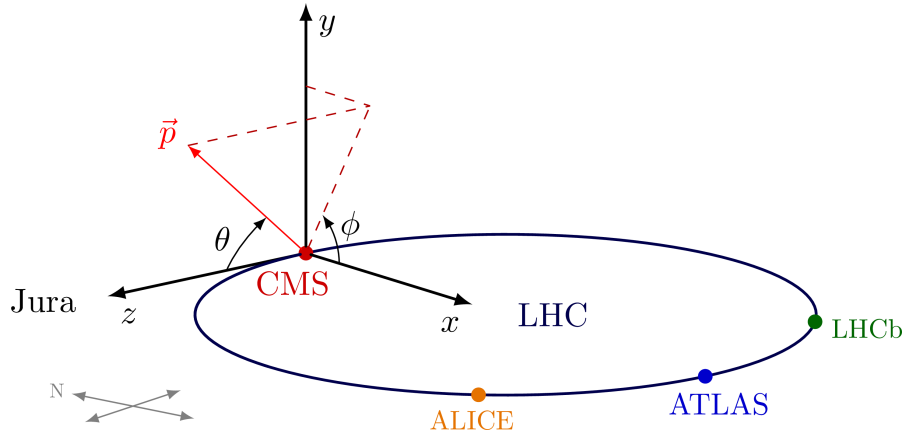



Figura 1.1: Esquema de la distribución de los distintos detectores en el LHC [1]

El diagrama completo del sistema de referencia usado en el CMS puede verse en la Figura 1.2.

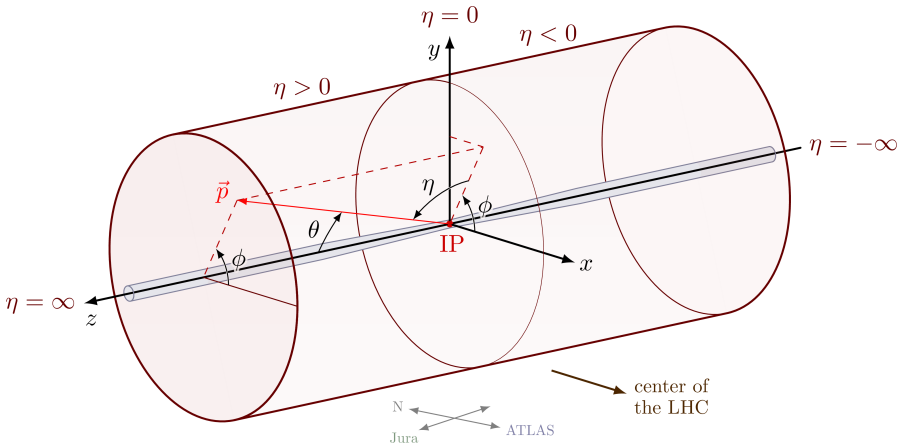



Figura 1.2: Sistema de referencia en detalle usado por el experimento CMS [1]

## 1.2. El modelo estándar y los distintos procesos físicos

A mediados de la segunda mitad del siglo veinte se empezó a desarrollar una teoría que terminó convirtiéndose en una de las más exitosas de toda la física. Esta teoría, conocida como el modelo estándar, terminaría de formularse a mediados de la década del setenta y desde entonces se ha convertido en el foco de estudio de cientos de científicos a lo largo del planeta. Si bien no es una teoría del todo y hay varios fenómenos físicos que no logra explicar, ha logrado predecir con éxito la existencia de partículas elementales como los bosones W y Z, así como también el famoso bosón de Higgs. Las confirmaciones experimentales de este tipo de predicciones se logran mediante los aceleradores de partículas. Como se mencionó previamente, el LHC colisiona haces de protones a muy altas velocidades. Es en estos choques en donde se produce la física realmente interesante y en dónde existe la posibilidad de encontrar nuevas partículas o campos fundamentales. Es de mucho interés en la actualidad buscar en los resultados de estas colisiones propiedades u objetos que el modelo estándar falle en predecir, pues eso indicaría que estamos más cerca de encontrar una teoría más completa. Lastimosamente todos los experimentos que se han realizado hasta la fecha han verificado el modelo estándar en vez de ponerlo contra las cuerdas, lo que lo convierte, como dijimos anteriormente, en una de las teorías mas robustas y clave de la física moderna. El modelo estándar actualmente describe tres de las cuatro fuerzas fundamentales del universo, y clasifica a todas las partículas elementales de una manera general en dos grandes grupos: fermiones y bosones fundamentales.

### 1.2.1. Fermiones fundamentales

Se denominan fermiones a aquellas partículas elementales que tienen un espín semi-entero, por ejemplo $\frac{1}{2}$ y que cumplen con el principio de exclusión de Pauli. Se dividen en dos grandes categorías, quarks (también escrito en español usualmente como cuarks) y leptones. Y en ambos casos cuentan con sus respectivas antipartículas.

#### Quarks

Existen seis tipos de quarks: u (arriba), d (abajo), c (encantado), s (extraño), t (cima), y b (fondo). Las letras corresponden a los nombres en inglés. Su característica principal es que tienen carga de color, la cual puede ser roja, verde, azul, o antirroja, antiverde y antiazul en el caso de los anti-quarks. Los quarks se encuentran siempre confinados en la naturaleza, por lo que es imposible que uno exista de manera individual. Cuando tres quarks se juntan forman un barión y cuando un quark y un anti-quark se juntan forman un mesón. En ambos casos el resultado es una partícula de color neutral (los tres colores juntos o un color y su respectivo anticolor) denominada hadrón. Los ejemplos más famosos de bariones son los protones y neutrones, que están hechos de quarks uud y udd respectivamente.

#### Leptones

Al igual que los quarks, existen seis tipos de leptones: electrones, muones, taus, neutrino tipo electrón, neutrino tipo muon y neutrino tipo tau. Los leptones no tienen carga de color.

### 1.2.2. Bosones fundamentales

Contrario a los fermiones los bosones son aquellas partículas que tienen espín entero y que no cumplen con el principio de exclusión de Pauli, por lo que no tienen antipartículas. Al igual que los fermiones podemos clasificarlos en dos categorías, bosones de gauge y bosones escalares.

#### Bosones de gauge

Los bosones actúan como portadores de fuerza, y son los responsables de llevar a cabo las interacciones fundamentales de la fuerza fuerte, débil y la electromagnética. Tienen espín uno y actualmente sabemos que existen cuatro de estos bosones: gluón, fotón, bosón Z y bosón W.

#### Bosones escalares

El único bosón escalar conocido es el bosón de Higgs. Tiene espín cero y cumple un rol fundamental para explicar porqué las partículas elementales tienen masa

(exceptuando los fotones y gluones). Fue teorizado en la década del sesenta por Peter Higgs y otros y no fue sino hasta el año 2012 que los experimentos CMS y ATLAS del CERN pudieron confirmar su existencia [6].

La clasificación de la que acabamos de hablar puede ser resumida de una manera simple en una única tabla como se muestra en la Figura 1.3

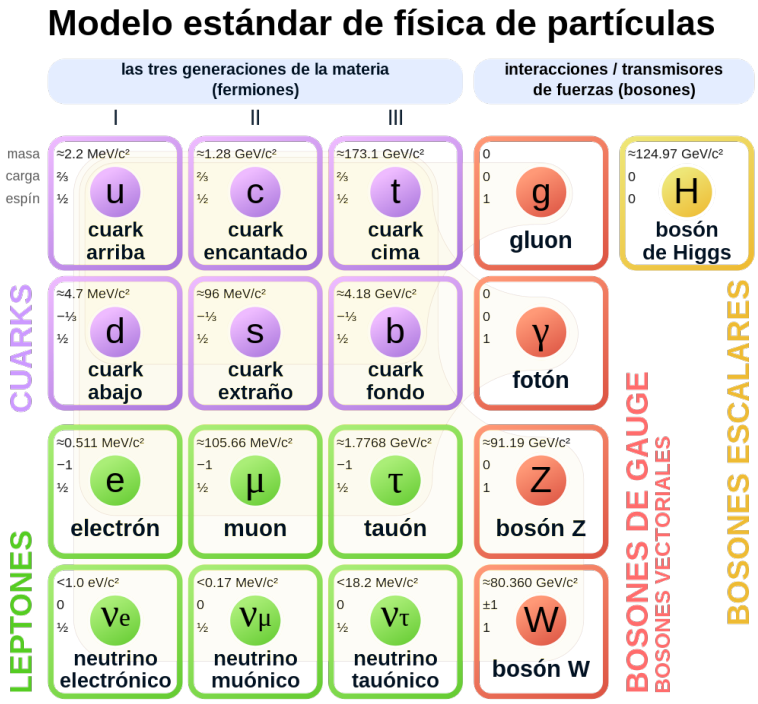



Figura 1.3: Clasificación de las partículas elementales del modelo estándar [2]

## 1.3. Muones

Los muones son partículas elementales con carga negativa, espín $\frac{1}{2}$ y tienen una masa de 105.66 $\frac{MeV}{c^2}$. Debido a su masa los muones emiten menor radiación

Bremsstrahlung, por lo que penetran mayor cantidad de materia, lo que los hace atravesar los calorímetros internos del CMS. Es por esto que su detección se logra a través de cuatro estaciones especializadas que están localizadas en la parte exterior del solenoide. Cada una de estas estaciones detecta y rastrea la posición del muon. Debido a la distancia que recorren y al campo magnético generado por el solenoide la trayectoria de los muones se curva de manera proporcional a su momento como se puede ver en la Figura 1.4.

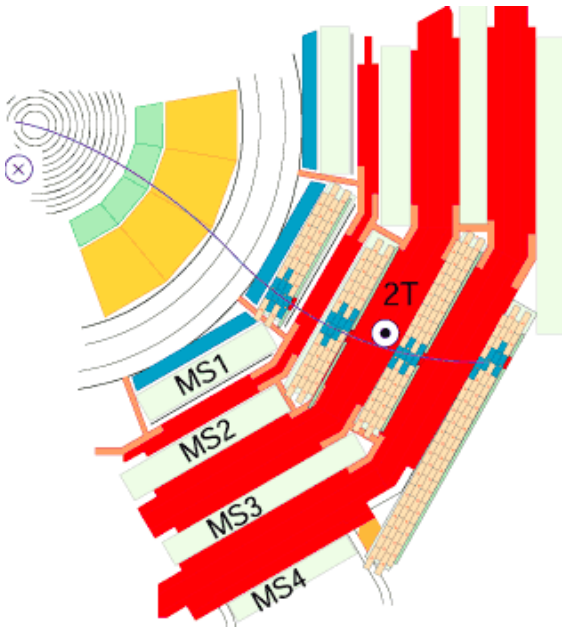



Figura 1.4: Diagrama de la detección de muones en el detector CMS [3]

## 1.4. Jets

Si bien nuestro enfoque principal estará en los muones como se explicará más adelante, es importante conocer otros objetos físicos como los jets. Se denomina

jet al resultado de una hadronización de un quark o un gluón luego de una colisión. Los jets están compuestos de distintas partículas en distintas proporciones energéticas, mayoritariamente hadrones cargados (65 %) y fotones (25 %), con un (10 %) restante de hadrones neutrales.

## 1.5. MET

MET, del inglés missing transverse energy (energía transversal perdida), se refiere a la magnitud de la suma negativa de los vectores de momento transversal de todas las partículas en un evento. Es especialmente útil para poder inferir la presencia de objetos que no dejan rastro en el detector, como por ejemplo neutrinos.

## 1.6. Diagramas de Feynman

Los diagramas de Feynman son representaciones gráficas propuestas por Richard Feynman en el año de 1948, que nos ayudan a entender el comportamiento y las interacciones de las partículas subatómicas. Gracias a ellos podemos calcular la probabilidad de que ciertas interacciones y decaimientos ocurran ya que nos ayudan a visualizar formulaciones matemáticas que de otra manera serían demasiado abstractas para tratar. Estas probabilidades asociadas a los distintos diagramas son de suma importancia para el análisis de física de partículas, ya que si sabemos la probabilidad de que ciertos procesos ocurran podemos comparar ese valor con el obtenido en colisiones reales. Es precisamente este procedimiento el que de manera

general se utiliza para analizar experimentos de física de partículas. De esta forma también es cómo se puede llegar a descubir física más allá del modelo estándar (BSM por sus siglas en inglés). Si observamos en datos de colisiones, por ejemplo, un exceso de eventos alrededor de alguna energía en particular, puede que esa resonancia se corresponda con un proceso de muy baja sección eficaz (un proceso exótico), sin embargo éste podría no ser el caso y tendríamos por lo tanto evidencia prometedora de física más allá del modelo estándar. Resulta útil, entonces, encontrar una manera de usar los diversos algoritmos de clasificación de imágenes para poder discriminar este tipo de resonancias. Para ejercitar esta técnica hemos escogido los procesos de Drell-Yan, $t\bar{t}$ + jets y W + jets.

### 1.6.1. Proceso Drell-Yan

El proceso Drell-Yan ocurre cuando un quark y un antiquark se aniquilan mutuamente y generan un bosón Z o un fotón virtual, éste posteriormente decae en un leptón y un antileptón. Su respectivo diagrama de Feynman se puede ver en la Figura 1.5. Este proceso nos resulta especialmente útil ya que presenta una resonancia muy parecida a la que podríamos esperar al momento de buscar física más allá del modelo estándar. Usaremos esta resonancia como nuestra señal principal.

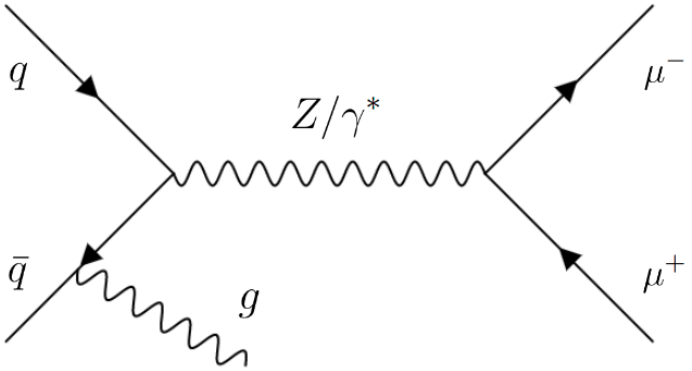



Figura 1.5: Diagrama de Feyman del proceso Drell-Yan

Podemos notar en la Figura 1.5 que existe la posibilidad de encontrar jets en este proceso gracias al glúon en la parte inferior izquierda.

### 1.6.2. Proceso $t\bar{t}$ (top y anti-top)

Usaremos los eventos del proceso de $t\bar{t}$ como señal de fondo. Podemos notar en su respectivo diagrama de Feynman en la Figura 1.6 que este proceso también nos permite llegar a un estado final de dos leptones con jets.

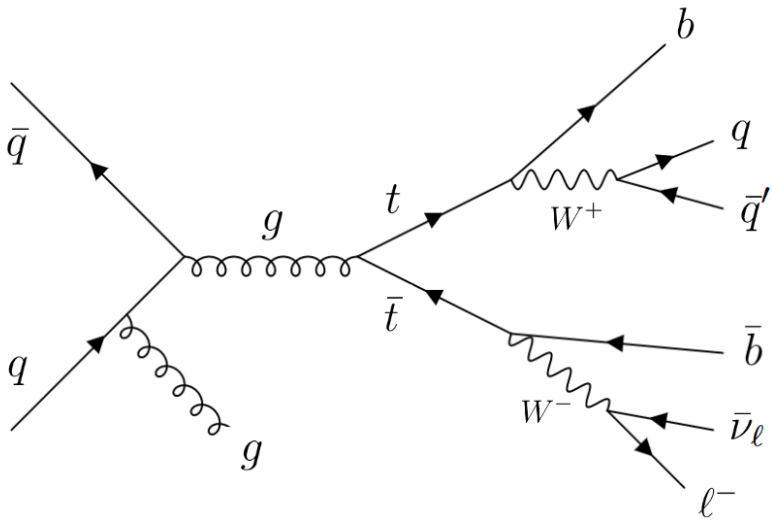



Figura 1.6: Diagrama de Feyman del decaimiento de un quark top y anti-top

### 1.6.3. Decaimiento de un bosón W

Al igual que con el proceso $t\bar{t}$, usaremos el decaimiento de los bosones W como señal de fondo gracias a que nuevamente nos permite llegar a un estado final de dos leptones como se muestra en su diagrama en la Figura 1.7.

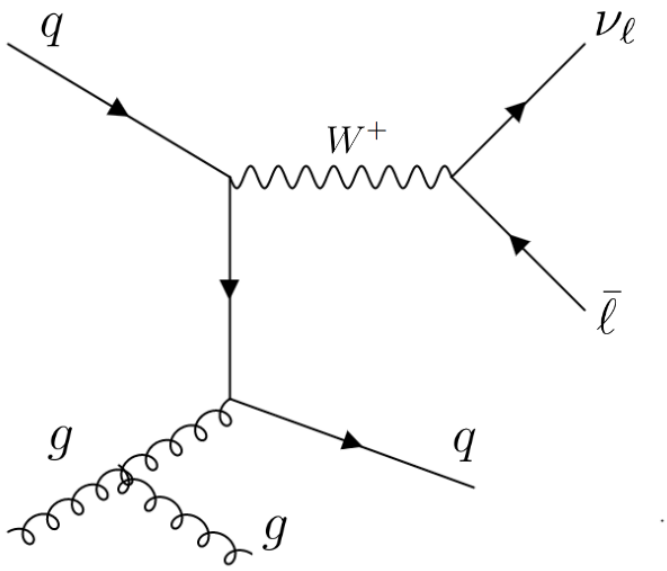



Figura 1.7: Diagrama de Feyman del decaimiento de un bosón $W^+$

Si bien podríamos usar cualquier leptón para el análisis hemos escogido usar muones ya que son los que se encuentran en mayor proporción en los decaimientos, además de que son de los objetos a los que más importancia se les da en el experimento CMS como bien indica su nombre.

## 1.7. Masa invariante

Podemos entender a la masa invariante como una cantidad de un sistema que no cambia dependiendo del observador. En relatividad especial, los sistemas de referencia juegan un papel importante al momento de calcular las propiedades físicas de los objetos, ya que no es lo mismo calcular, por ejemplo, el momento

de una partícula en una colisión visto desde la propia partícula, a calcularlo en el sistema del laboratorio. Para definir formalmente la masa invariante definiremos antes el cuadrimomento $P^\mu$ y la cuadrivelocidad $U^\mu$.

$$(U^\mu) = (\gamma c, \gamma \vec{v}) \tag{1.2}$$

Donde $\gamma$ es el factor de Lorentz que se define como $\gamma = \frac{1}{1-\beta^2}$ con $\beta = \frac{v}{c}$, $c$ es la velocidad de la luz y $\vec{v}$ es la velocidad de la partícula medida desde un sistema de referencia inercial. Definimos el cuadrimomento de la siguiente manera

$$(P^\mu) = \left(\frac{E}{c}, \vec{P}\right) \tag{1.3}$$

$$P^\mu = m_o U^\mu \tag{1.4}$$

Donde $\vec{P}$ es el momento en tres dimensiones. Calculando $P^\mu P_\mu$ y haciendo $c = 1$, como es común en física de partículas, tenemos

$$P^\mu P_\mu = m_o^2 = E^2 - |\vec{P}|^2 \tag{1.5}$$

Donde $m_o$ es precisamente la masa invariante que buscábamos. Podemos entonces calcular la masa invariante de una colisión de dos partículas de la siguiente forma

$$M^2 = (E_1 + E_2)^2 - (\vec{P_1} - \vec{P_2})^2 \tag{1.6}$$

Donde los subíndices 1 y 2 indican respectivamente el momento y energía de la partícula correspondiente. Vamos a trabajar con las coordenadas que se usan en los

aceleradores. En concreto, buscamos llegar a una expresión de la masa invariante en función de $\eta$ y $\phi$. Para esto usaremos las siguientes expresiones del momento y energía

$$\vec{P} = P_T \cos\phi\hat{i} + P_T \sin\phi\hat{j} + P_T \sinh\eta\hat{k} \tag{1.7}$$

$$|\vec{P}| = P_T \cosh\eta \tag{1.8}$$

$$E = \sqrt{P_T^2 + m^2}\cosh y \tag{1.9}$$

Donde $y$ es la rapidez convencional definida como:

$$y = \frac{1}{2}\ln\left(\frac{E + P_z}{E - P_z}\right) \tag{1.10}$$

y $P_T = \sqrt{P_x^2 + P_y^2}$ el momento transversal. Ya que las partículas en los colisionadores alcanzan velocidades cercanas a las de la luz, podemos despreciar la masa de la mayoría de leptones respecto de su momento. Es decir $m \ll |\vec{P}|$, por lo tanto usando la ecuación 1.5 tenemos que $E \approx |\vec{P}|$ además de poder aproximar $\eta$ a $y$. Tenemos entonces que la ecuación 1.9 se convierte en

$$E = P_T \cosh\eta \tag{1.11}$$

Expandiendo la expresión de la masa invariante de la ecuación 1.6 tenemos

$$M^2 = E_1^2 + E_2^2 + 2E_1E_2 - 2\vec{P_1}\cdot\vec{P_2} - |P_1|^2 - |P_2|^2 \tag{1.12}$$

$$M^2 = 2E_1E_2 - 2\vec{P_1}\cdot\vec{P_2}. \tag{1.13}$$

Calculamos el producto punto

$$\vec{P_1} \cdot \vec{P_2} = P_{T1}P_{T2}(\cos\phi_1 \cos\phi_2 + \sin\phi_1 \sin\phi_2 + \sinh\eta_1 \sinh\eta_2) \tag{1.14}$$

$$\vec{P_1} \cdot \vec{P_2} = P_{T1}P_{T2}(\cos(\phi_1 - \phi_2) + \sinh\eta_1 \sinh\eta_2). \tag{1.15}$$

Reemplazando en el ecuación 1.11

$$M^2 = 2P_{T1}\cosh\eta_1 P_{T2}\cosh\eta_2 - 2\left(P_{T1}P_{T2}(\cos(\phi_1 - \phi_2) + \sinh\eta_1 \sinh\eta_2)\right) \tag{1.16}$$

Con lo que finalmente tenemos

$$M^2 = 2P_{T1}P_{T2}\left(\cosh(\eta_1 - \eta_2) - \cos(\phi_1 - \phi_2)\right) \tag{1.17}$$

Esta ecuación es especialmente útil para física de colisiones.

# Capítulo 2

# Métodos

Cómo se discutió en la sección anterior nuestro objetivo será extraer información de los procesos escogidos y poder usarla para entrenar un modelo de una red neuronal convolucional. Las bases de datos necesarias las encontraremos en el portal de datos abiertos del CERN y la manera de extraer esta información será en gran medida aportada por el taller público que ofrece el CMS. Antes de continuar con los detalles es importante mostrar que el código usado en este trabajo puede ser encontrado en el siguiente repositorio: `https://github.com/jose8af/cnn-hep-thesis` [7].

## 2.1. Datos abiertos: CERN Open Data portal

Gracias a la popularidad y a la relevancia del CERN, la organización cuenta con una política de datos abiertos. A través de la web CERN Open Data portal, cualquier persona puede acceder tanto a datos de simulaciones como a datos de colisiones reales de las distintas colaboraciones. Nosotros nos enfocaremos en los datos abiertos del experimento CMS [8]. Hacer uso de estos datos no es necesariamente fácil, y afortunadamente la colaboración CMS saca anualmente un taller abierto al público para poder comprender y usar de manera correcta los datos abiertos. Existe una cantidad increíblemente amplia de datos disponibles; los que nos interesan son los aquellos obtenidos de la corrida dos (RUN 2) del año 2015, los archivos se encuentran en formato miniAOD y miniAODSIM. En concreto, haremos uso de 4 datasets que se detallan en el Cuadro 2.1.

| **Tipo** | **Nombre** | **Tamaño [TB]** | **Rec ID** |
|---|---|---|---|
| Colisión | **/DoubleMuon/Run2015D-16Dec2015-v1/MINIAOD** | 0.86 | 24127 |
| Simulación | **TT_TuneCUETP8M1_13TeV-powheg-pythia8** | 3.4 | 19980 |
| Simulación | **WJetsToLNu_TuneCUETP8M1_13TeV-amcatnloFXFX-pythia8** | 3.8 | 20548 |
| Simulación | **ZToMuMu_M_50_120_NNPDF30_13TeV_powheg_herwigpp** | 0.06 | 21579 |

Cuadro 2.1: Datasets usados en el presente trabajo

### 2.1.1. CMSSW

CMSSW [9] es un software escrito en C++. Consta de una colección muy amplia de librerías que se especializan en adquirir, procesar y analizar datos del experimento CMS. A pesar de que está escrito en C++, la configuración se la realiza mediante scripts de python.

### 2.1.2. POET

POET, acrónimo en inglés para Physics Object Extractor Tool (Herramienta para la extracción de objetos físicos) [10], es una herramienta que fue desarrollada en un inicio como algo pedagógico que sirviera para acceder y extraer la información de los objetos físicos de una colisión. La idea general de POET radica en tratar a los distintos objetos de manera individual. Esto lo hace a través de Analyzers, módulos escritos en C++ que nos permiten acceder a los distintos eventos de un dataset. Toda la información que queramos extraer usando POET nos será arrojada en un archivo de extensión .root.

### 2.1.3. Partículas generadoras

Una pregunta útil al momento de trabajar con simulaciones es cómo se obtienen estos datos. En el caso de las simulaciones del CMS todo parte de eventos generadores, también llamado verdad MC (Monte Carlo) o Nivel de verdad (originalmente en inglés es “MC Truth” o “Truth level”). Este nivel de verdad contiene

la información más pura detrás de los objetos en un evento simulado. Es por lo tanto importante poder acceder a este tipo de información. Para hacerlo es necesario realizar un "matching" (emparejamiento) entre los muones reconstruidos, es decir los muones que vemos en nuestro archivo root, y los muones a nivel de generador. Para acceder a los muones a nivel de generador es necesario saber su identificador. Todas las partículas tiene un código llamado PdgID (del ingés Particle Data Group Identification) [11], en el caso de los muones el número correspondiente es el 13. El emparejamiento se realizará calculando la mínima distancia angular $\Delta R$ entre los muones reconstruidos y los generadores. La distancia angular nos dice qué tanto dos objetos se están moviendo en la misma dirección y se define de la siguiente forma:

$$\Delta R = \sqrt{(\Delta\eta)^2 + (\Delta\phi)^2} \tag{2.1}$$

Tras realizar el emparejamiento lo que tendremos será para cada muon reconstruido un único muon de nivel generador tal que su distancia angular sea mínima, podemos guardar la información de este muon a nivel generador y usarla para verificar propiedades del dataset que de otra forma estarían escondidas. Una de estas propiedades que podemos verificar, al menos en los datasets de los procesos Drell-Yan, es la posición del pico de la resonancia del bosón Z. Aunque esto pueda parecer redundante no lo es, ya que existen diversos datasets de procesos Drell-Yan que tienen una masa invariante desplazada de los 91 GeV esperados. La Figura 2.1 corresponde a la masa invariante de los muones a nivel de generador del dataset ZtoMuMu.

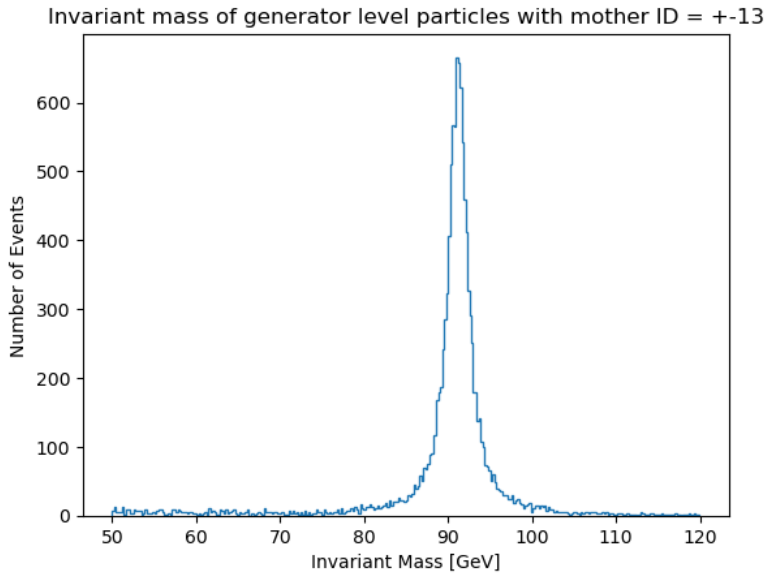



Figura 2.1: Masa invariante de partículas generadoras con mother ID = ± 13

Es notorio que el pico de la resonancia está perfectamente centrado en 91 GeV. Y por lo tanto al graficar la masa invariante usando los muones reconstruidos se ve el mismo pico en la misma energía, que es lo esperado. La Figura 2.2 muestra la masa invariante calculada usando los muones reconstruidos.

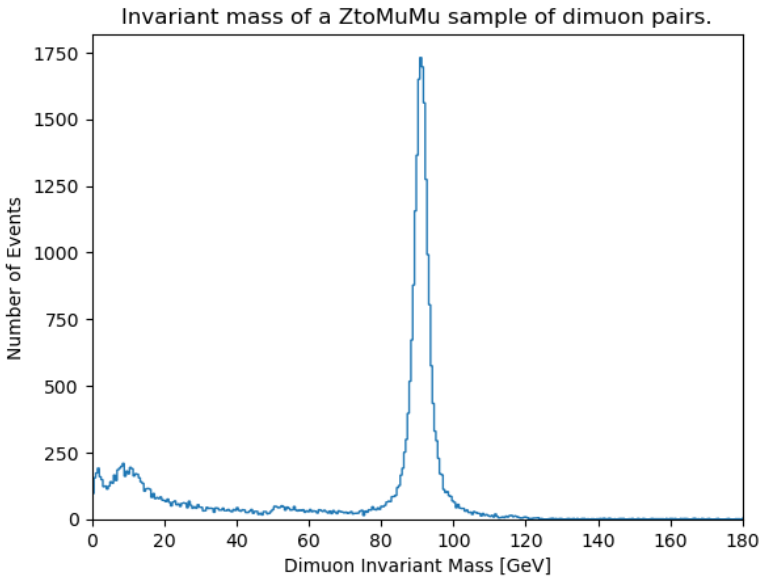



Figura 2.2: Masa invariante de dimuones reconstruidos pertenecientes al dataset ZtoMuMu

## 2.2. Representación gráfica de eventos

Si lo que queremos es usar un modelo de clasificación de imágenes, lo primero que tendremos que hacer, es encontrar una manera de representar las variables más importantes como $P_T$, $\eta$ y $\phi$ de los distintos objetos de una manera visual y sencilla. Usaremos la propuesta de Fernández et. al. [12], que involucra centrar a los objetos de acuerdo a su posición en el acelerador y dibujarlos con un radio proporcional a su momento transversal siguiendo la siguiente fórmula

$$R = \alpha \cdot \ln p_T. \tag{2.2}$$

Donde $\alpha$ es una constante. En el caso del MET, simplemente dejaremos el valor de $\eta = 0$, por lo que siempre estará centrado en esa dirección. Los objetos así mismo estarán diferenciados a través del color, siendo los muones rojos, los jets azules y el MET negro. Para poder realizar una buena representación y obtener imágenes útiles es necesario tener en cuenta ciertos aspectos. Primero, debido a la naturaleza de la red todas nuestras imágenes deben ser de 244×244 píxeles. También es necesario prevenir el solapamiento entre objetos y que los mismos, debido a la escala, no se salgan de los bordes establecidos. Todos estos problemas se solucionan con distintos factores de escala que deberán ser escogidos de manera manual de acuerdo al evento de mayor energía. Si nos aseguramos que éste permanezca dentro de los bordes de nuestra imagen automáticamente el resto de eventos también estarán confinados.

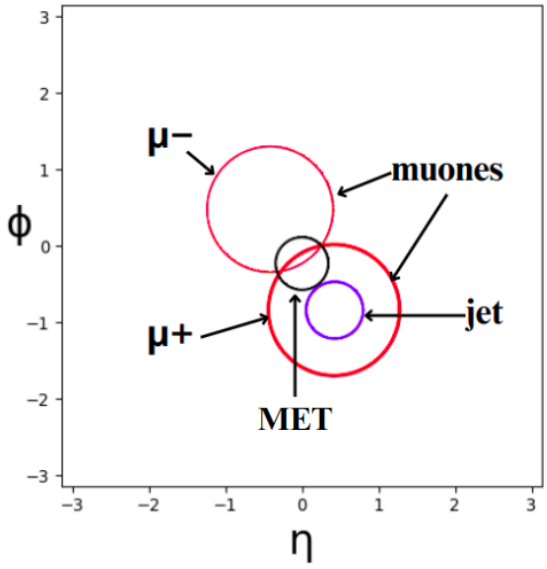
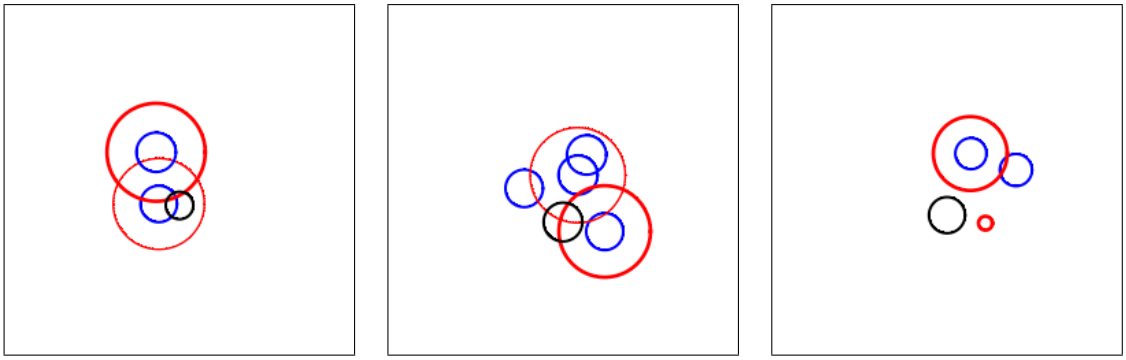



Figura 2.3: Diagrama de la representación visual de un evento

Una carácteristica implementada en la representación fue la visualización de

la carga de los muones. Como se puede ver en la Figura 2.3 la línea más delgada representa a los muones con carga negativa y la línea gruesa representa a los muones positivos. La Figura 2.4 muestra imágenes ejemplo correspondientes a los diferentes decaimientos.



(a) Drell-Yan (b) $t\bar{t}$ + jets (c) W + jets

Figura 2.4: Representación gráfica de diferentes eventos correspondientes a los diferentes procesos físicos

## 2.3. Redes neuronales

Desde el nacimiento de la informática una de las principales preguntas que surgieron es si las máquinas podrían imitar de alguna forma el cerebro humano. Esto implicaba, entre otras cosas, la capacidad de aprendizaje, el poder adquirir conocimiento en base a diversos datos. Esto plantea el inicio de la inteligencia artificial y posteriormente las redes neuronales, algoritmos que pudiesen de alguna manera replicar las capacidades humanas. A pesar de que las bases teóricas e incluso prácticas de estos algoritmos llevan existiendo desde hace décadas, no ha sido sino hasta estos últimos años que estas ideas han cobrado verdadera fuer-

za, desarrollándose a velocidades vertiginosas y produciendo programas capaces de pintar, escribir, diseñar, entre otras múltiples actividades que en un inicio se pensaba eran únicas de los seres humanos. Una red neuronal, como bien lo dice su nombre, funciona con unidades fundamentales llamadas neuronas, éstas neuronas se encuentran agrupadas en capas y existen tres tipos de capas fundamentales: capa de entrada, capas ocultas y capa de salida. Todas las capas pueden conectarse entre sí a través de sus respectivas neuronas. Podemos ver un diagrama de las distintas capas en la Figura 2.5

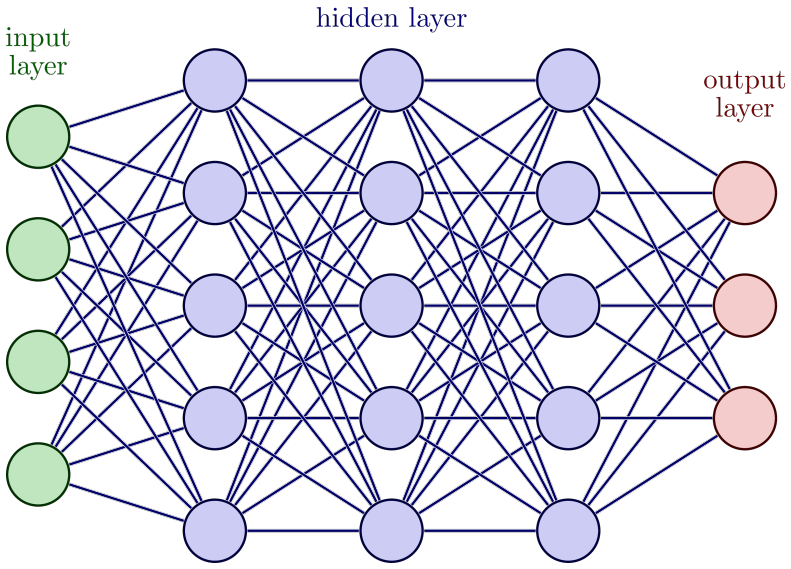



Figura 2.5: Diagrama de las capas de una red neuronal [4]

### 2.3.1. Redes neuronales convolucionales

Las redes neuronales convolucionales (CNN por sus siglas en inglés) son una subclase de las redes neuronales y tienen como mínimo una capa de convolución. Además los datos de entrada tendrán una estructura diferente a los de una red

neuronal convencional, siendo estos matrices tridimensionales. Este tipo de redes representa una ventaja respecto a las redes neuronales tradicionales ya que reduce drásticamente el número de parámetros de la red, así como también la complejidad de la misma, esto es de especial utilidad al momento de trabajar con imágenes, cuya representación visual puede ser expresada como una matriz en tres dimensiones. La Figura 2.6 muestra un diagrama de las capas en una red neuronal convolucional.

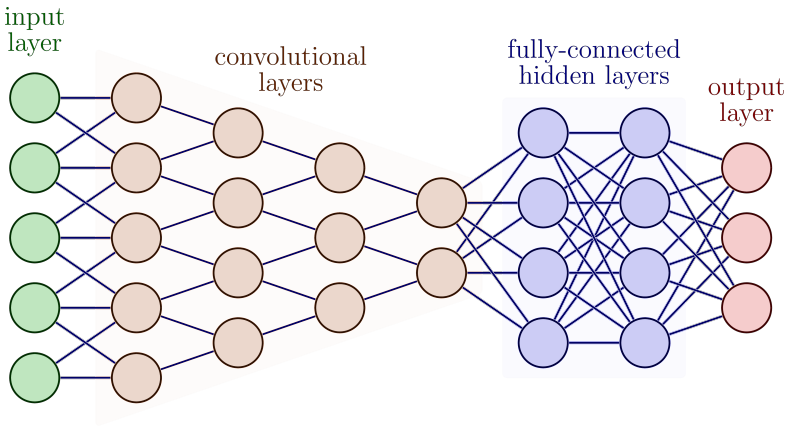



Figura 2.6: Diagrama de las capas de una red neuronal convolucional [4]

Explicaremos brevemente una serie de conceptos y librerías importantes al momento de crear y entrenar un modelo de red neuronal convolucional.

- Conjunto de entrenamiento: Es aquel conjunto de imágenes destinado únicamente a entrenar a nuestro modelo. Es decir, los pesos de cada neurona se irán actualizando cada vez que el modelo recorra estos datos.

- Conjunto de prueba: Este conjunto de imágenes está destinado a evaluar el rendimiento del modelo en datos que nunca antes había visto. Este conjunto de datos no interactúa nunca con el modelo sino hasta que éste ya está

entrenado, es decir, no puede alterar los pesos de las neuronas.

- Conjunto de validación: El conjunto de imágenes de validación sirve para ir monitoreando el rendimiento del modelo en datos que nunca ha visto. Esto lo hace varias veces a lo largo del entrenamiento.

- Épocas (Epochs): Es el número de veces que el modelo recorrerá el conjunto de entrenamiento. Es por lo tanto una manera de ajustar el tiempo de duración del entrenamiento.

- Tamaño del lote (Batch size): Es el número de imágenes que tomará el modelo al momento de entrenarse y actualizar sus pesos. Mientras menor sea este número más lento será el entrenamiento pero a su vez se ajustarán los pesos de las neuronas un mayor número de veces.

- TensorFlow: Tensorflow [13] es una librería de código abierto que ha ganado popularidad especialmente en el desarrollo de aplicaciones de machine learning .

- Keras: Keras [14] es una librería escrita en Python capaz de correr encima de Tensorflow. Actualmente es una de las API más usadas para desarrollar redes neuronales.

- Precisión: Es una de las métricas más útiles al momento de evaluar nuestro modelo. Se define como el número de predicciones correctas dividido el número total de predicciones.

- Función de pérdida: La función de pérdida nos ayuda a evaluar qué tan bien se desempeña nuestro modelo. Si una predicción se desvía demasiado

entonces el valor de la función de pérdida será alto. Querrémos por tanto siempre minimizar esta función.

- Función de activación: La función de activación será la encargada de escoger la señal de salida de una neurona y de alguna forma convertirla en una señal del entrada que pueda ser usada por por las siguientes neuronas. Es en este proceso que se ejecuta el aprendizaje del modelo. Entre las funciones de activación más usadas se encuentra ReLu [15], definida como $f(x) = max(0, x)$ y softmax [16], la cual produce valores acotados entre 0 y 1 por lo que es usada en la última capa en los modelos de clasificación.

- Sobreajuste (overfitting): Se dice que un modelo está sobreajustado si su rendimiento en los datos de entrenamiento es perfecto. Esto le impide funcionar correctamente en datos que no conoce. El sobreajuste se da cuando un modelo es entrenado por demasiado tiempo en un conjunto de datos, ya que empieza a aprenderse patrones específicos de aquel conjunto y pierde generalidad. Existen varias maneras de prevenir el sobreajuste, estas pueden ser añadir penitencias aleatorias a los pesos de las neuronas, detener el entrenamiento si el modelo ya no mejora su precisión con el pasar de las épocas en un conjunto de datos de validación, o reducir la tasa de aprendizaje.

### 2.3.2. Bases de datos

Usando los datasets del Cuadro 2.1 se generaron un total de 83097 imágenes, en donde cada imagen corresponde a un evento, es decir, a una colisión. Por un motivo netamente de recursos y tiempo, se decidió fraccionar a la base de datos

completa y armar a partir de ella una base de datos reducida con 32904 imágenes, esto con el fin de poder realizar la mayor cantidad de variaciones posibles en el modelo y evitar tiempos de entrenamiento demasiado prolongados. Para las predicciones se usó un conjunto de 12722 imágenes provenientes del set de datos en [17]. Como es normal en este tipo de entrenamientos con redes convolucionales, separamos nuestra base de datos en tres categorías estándar: Entrenamiento, prueba y validación. Separados en una proporción de 60 | 20 | 20 para la base de datos reducida y 70 | 10 | 20 para la base de datos completa. Los números exactos de los respectivos conjuntos se pueden ver en los Cuadros 2.2 y 2.3.

| **Clase** | **Entrenamiento** | **Validación** | **Prueba** |
|---|---|---|---|
| $t\bar{t}$ + jets | 6580 | 2193 | 2195 |
| Drell-Yan | 6580 | 2193 | 2195 |
| W + jets | 6580 | 2193 | 2195 |
| **Total** | 19740 | 6579 | 6585 |

Cuadro 2.2: Número de imágenes usadas en los conjuntos de entrenamiento, validación y prueba en la base de datos reducida

| **Clase** | **Entrenamiento** | **Validación** | **Prueba** |
|---|---|---|---|
| $t\bar{t}$ + jets | 19389 | 5539 | 2771 |
| Drell-Yan | 19389 | 5539 | 2771 |
| W + jets | 19389 | 5539 | 2771 |
| **Total** | 58167 | 16617 | 8313 |

Cuadro 2.3: Número de imágenes usadas en los conjuntos de entrenamiento, validación y prueba en la base de datos completa

### 2.3.3. Preprocesamiento de los datos

Una de las maneras más comunes de aumentar nuestra base de datos en caso de no tener suficiente información es a través de las transformaciones. Una transformación es un proceso por el cual una imagen se verá modificada de diferentes formas con el fin de obtener una nueva imagen que nos sea útil para entrenar el modelo. Este tipo de transformaciones pueden ir desde modificaciones de la escala de color hasta rotaciones y traslaciones. En nuestro caso hemos usado cuatro tipos diferentes de técnicas que suelen ser estándar. Estas son:

- shear range: En español rango de corte. Distorsiona la imagen para obtener nuevas perspectivas.
- zoom range: En español rango de aumento. Realiza acercamientos a la imagen de manera aleatoria.
- Vertical flip: En español giro vertical. Voltea una imagen alrededor del eje-x.
- horizontal flip: En español giro horizontal. Voltea una imagen alrededor del eje-y.

El preprocesamiento de imágenes se hizo mediante la función de keras "preprocess_input". Esta función simplemente adecuará nuestras imágenes a un formato indicado para el modelo.

### 2.3.4. Arquitectura del modelo

La base de nuestro modelo será ResNet50 [18], esta es una red neuronal convolucional con 50 capas de profundidad. Podemos hacer uso de su versión preentrenada con los datos de ImageNet [19], esto con el fin de tener una arquitectura sólida que sepamos con anterioridad funciona correctamente. La capa exterior del modelo base fue removida y se añadió en su lugar una capa de "Global Average Pooling 2D"[20], esta capa nos ayuda a reducir drásticamente la cantidad de parámetros entrenables en la red. Se añadió también una capa densa con 1024 neuronas y activación ReLu y finalmente una capa densa con activación softmax que nos ayudará con las predicciones de clases.

### 2.3.5. Fine-Tuning

Una parte importante al momento de entrenar una red que tiene pesos establecidos por defecto es realizar ajustes finos (del inglés fine-tuning). Este proceso implica descongelar las capas superiores de nuestro modelo, en nuestro caso las diez últimas capas, y hacerlas entrenables, es decir que sus pesos puedan ajustarse, esto con el fin de que el modelo se adapte mejor a nuestra base de datos.

### 2.3.6. Compilación del modelo y entrenamiento

El modelo fue compilado con el optimizador Adam [21] y la función de pérdida escogida fue "categorical cross-entropy" [22]. El rendimiento del modelo fue

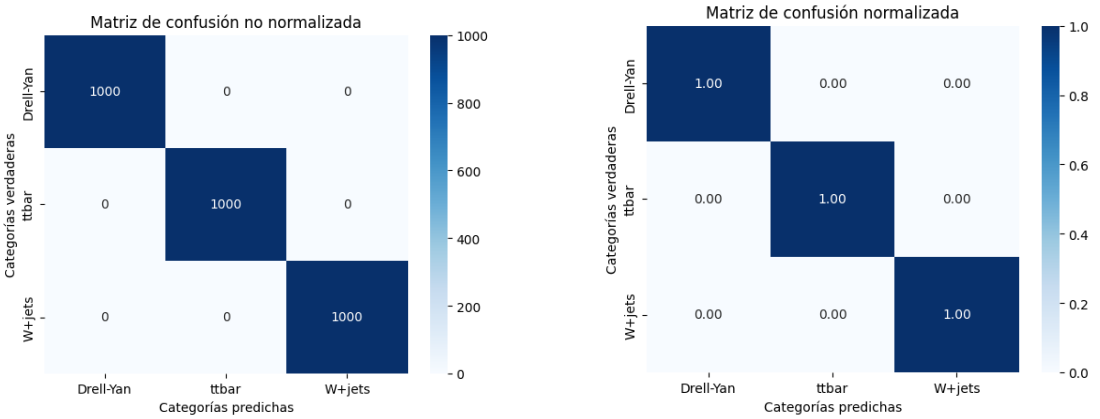
evaluado usando la precisión como métrica principal. Para el entrenamiento se estableció hacerlo con 20 o 40 épocas y lotes de 64 imágenes. Se añadió una reducción progresiva de la tasa de aprendizaje y una detención temprana para prevenir el sobreajuste.

### 2.3.7. Matrices de confusión

Antes de pasar a la sección de resultados es importante entender qué es y cómo leer e interpretar una matriz de confusión. Este tipo de matrices se han convertido en el estándar al momento de presentar resultados relacionados a los modelos de clasificación. Las matrices son cuadradas, y tienen como número de columnas o filas el número de categorías que el modelo está entrenado para clasificar. Dependiendo de la persona que las diseña, las columnas pueden representar las categorías predichas y las filas las categorías verdaderas o visceversa, y cada cuadrante representará el número de veces que el modelo predice determinada categoría. Por ejemplo, asumamos que hemos ya entrenado el modelo descrito en este trabajo y queremos comprobar su rendimiento. Para ello supondremos que lo hemos corrido sobre 3000 datos de prueba correspondientes de manera equitativa a las categorías Drell-Yan, $t\bar{t}$ y W+jets. Entonces tendremos una matriz 3×3 y una serie de números en los diferentes cuadrantes. Una matriz de confusión perfecta, es decir una perteneciente a los resultados de un modelo que nunca se equivoca al clasificar, tendrá en la diagonal los números 1000 y el resto de la matriz estará completado con ceros.

Usualmente es mucho más útil saber los porcentajes en comparación al número

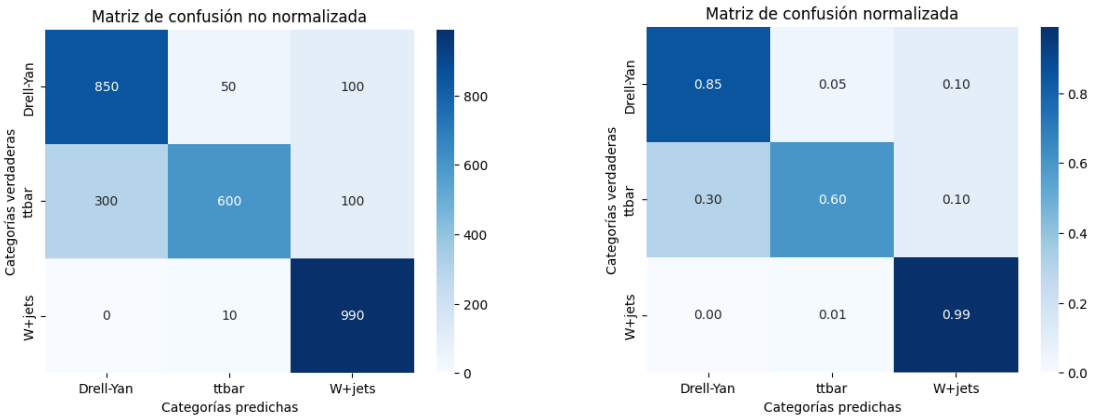
exacto de predicciones, por lo que es común dividir cada columna para el total de datos de prueba correspondiente. A la matriz resultante se la llama matriz normalizada, y tiene la ventaja de que es más fácil de interpretar ya que no necesitamos saber el número total de los datos usados para las pruebas.




(a) Matriz de confusión no normalizada (b) Matriz de confusión normalizada

Figura 2.7: Ejemplo de matrices de confusión de un modelo perfecto

La Figura 2.7 muestra como se verían las matrices de confusión de un modelo perfecto, por lo que lo que buscaremos será acercarnos tanto como sea posible a estas matrices. Desde luego, una matriz de un modelo verdadero no suele verse así. A continuación damos otro ejemplo con valores más dispersos.




(a) Matriz de confusión no normalizada (b) Matriz de confusión normalizada

Figura 2.8: Ejemplo de matrices de confusión

Analicemos rápidamente las matrices de la Figura 2.8. En este ejemplo con valores aleatorios, de los 1000 eventos Drell-Yan el modelo predijo 850 de manera correcta, 50 los categorizó como $t\bar{t}$ y 100 como W+jets. En el caso de los $t\bar{t}$, predijo 600 correctamente, confundió 300 con Drell-Yan y 100 con W+jets. Finalmente, de los 1000 eventos de W+jets, el modelo predijo 990 de manera correcta, no predijo ninguno como Drell-Yan y sólo categorizó de mala manera a 10 eventos como $t\bar{t}$.

# Capítulo 3

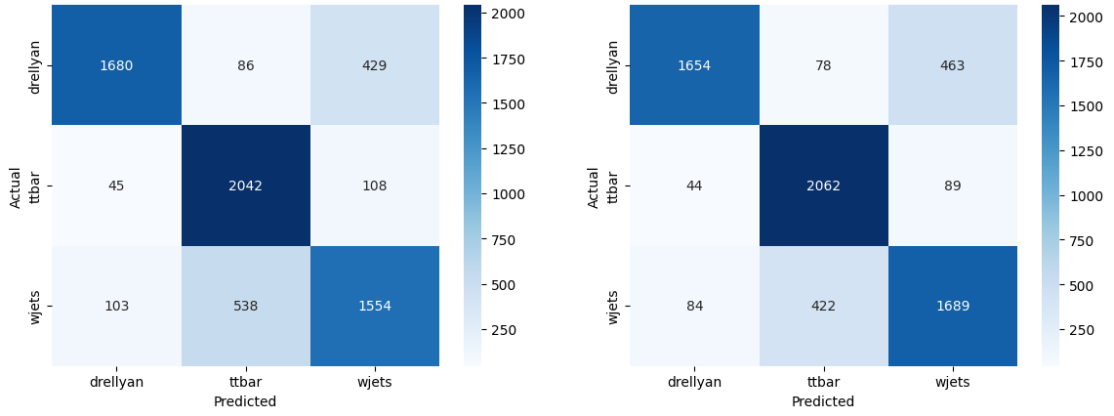
# Resultados

Debido a la falta de recursos GPU, los entrenamientos locales se hicieron usando un procesador intel CORE i7 de octava generación. Anticipando que los entrenamientos con la base de datos completa tardarían demasiado tiempo, se decidió hacer las pruebas principales con la base de datos reducida y una vez obtenida la información necesaria para saber cuál era el modelo con el tipo de imágenes más prometedoras, entonces se haría uso de recursos en línea como Google Colab [23] para entrenar la red con la base de datos completa. La primera prueba fundamental era saber la relación entre el número de jets presentes en las imágenes y el rendimiento de la red. Previamente se hicieron pruebas usando imágenes que tuvieran únicamente dos muones, sin embargo la precisión de la red era menor (menos del 60 %). Por lo tanto se partió con imágenes que contengan como mínimo un jet y la información correspondiente de MET. A continuación se presentan las precisiones y valores de pérdida encontrados para cada caso, así como las matrices

de confusión correspondientes.

| Número de jets | Precisión [%] | Valor de pérdida |
|---|---|---|
| 1 | 0.8012 | 0.4508 |
| 2 | 0.8208 | 0.4185 |
| 3 | 0.8355 | 0.3875 |
| 4 | 0.8416 | 0.3681 |

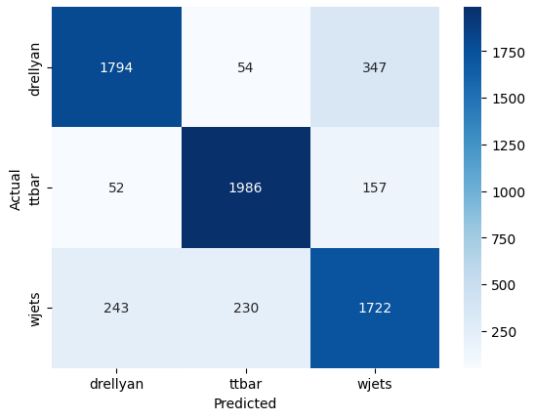
Cuadro 3.1: Precisión y valor de pérdida del modelo de acuerdo al máximo número de jets presentes en las imágenes



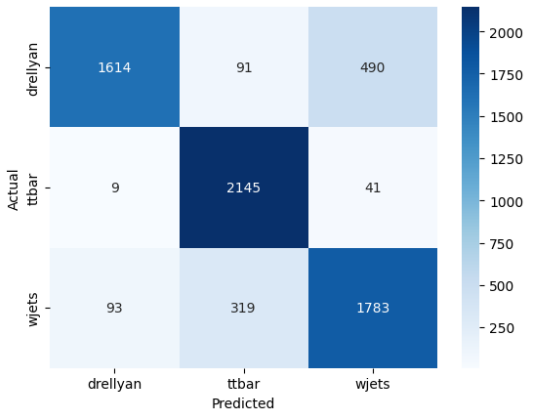

(a) 1 jet




(b) 2 jets

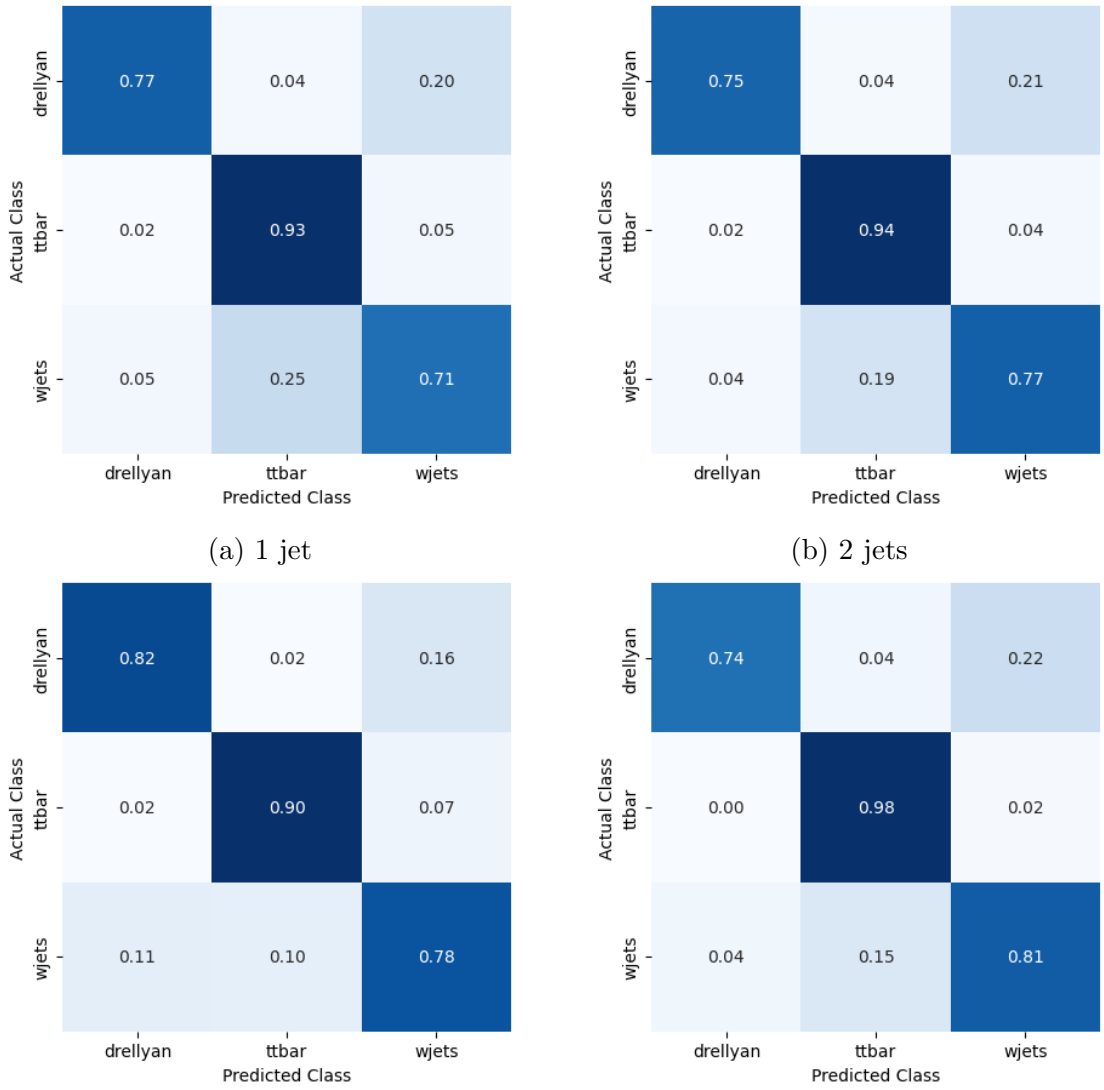



(c) 3 jets




(d) 4 jets

Figura 3.1: Matrices de confusión no normalizadas para distintos números de jets




(a) 1 jet

(b) 2 jets

(c) 3 jets

(d) 4 jets

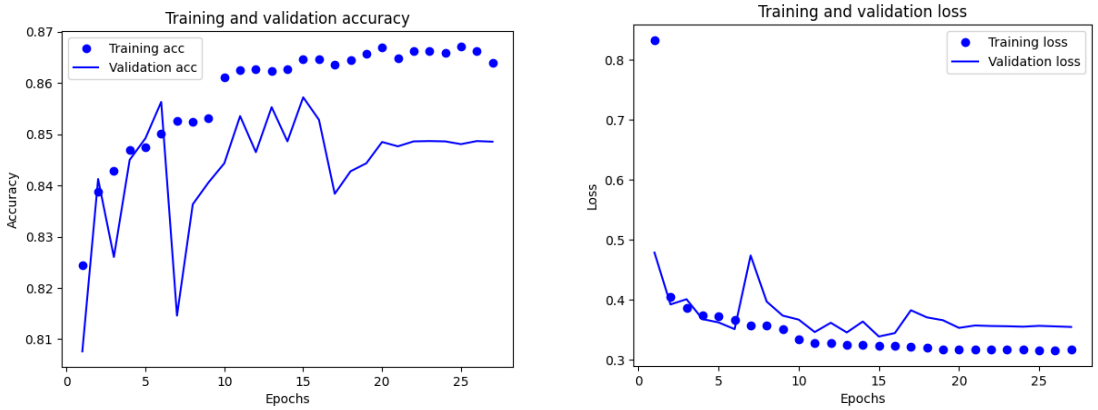
Figura 3.2: Matrices de confusión normalizadas para distintos números de jets

Lo primero que notamos es que el modelo es excepcionalmente bueno al momento de clasificar $t\bar{t}$, con más del 90 % de precisión en todos los casos. Lo siguiente que es visible es que de manera general el modelo tiene dificultades para distinguir entre los procesos Drell-Yan y W+jets. También tiene problemas para distinguir

entre $t\bar{t}$ y W+jets. Es también notorio, gracias al Cuadro 3.1, que a mayor número de jets mayor es la precisión de nuestro modelo. Podríamos por lo tanto seguir aumentando información en forma de jets pero esto causaría una pérdida de generalidad en el modelo lo que lo haría inútil para predicciones sobre datos reales. Como se discutió previamente, se hicieron ligeras variaciones del mejor modelo, en este caso el modelo entrenado con 4 jets con el fin de obtener la mayor precisión posible. A continuación se presentan los resultados del modelo que mejor desempeño tuvo entrenado en la base de datos reducida que denotaremos como B y en la base de datos completa que denotaremos como A. Estos últimos entrenamientos se realizaron en Google Colab, el cual es un servicio de Google pagado (aunque es posible usarlo gratuitamente con ciertos limitantes) que permite a cualquier usuario correr código en línea accediendo a recursos de hardware más potentes de los que usualmente tiene un computador convencional. Esto fue de especial utilidad al momento de entrenar la red con la base de datos completa, ya que redujo el tiempo de entrenamiento a 6 horas, lo que representó una mejora de casi 5 veces comparándolo con los entrenamientos usando recursos locales.

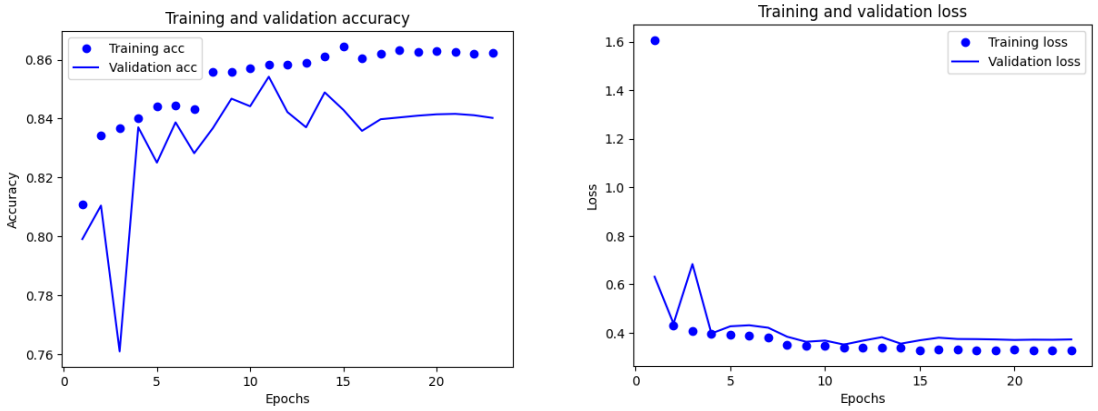
| **Tipo de base de datos** | **Precisión [%]** | **Valor de Pérdida** |
|---|---|---|
| Base de datos A | 0.8501 | 0.3532 |
| Base de datos B | 0.8524 | 0.3585 |

Cuadro 3.2: Precisión y valor de pérdida del modelo final entrenado en las distintas bases de datos




(a) Precisión (b) Valor de pérdida

Figura 3.3: Precisión y valor de pérdida del conjunto de validación y entrenamiento en el modelo final entrenado con la base de datos A

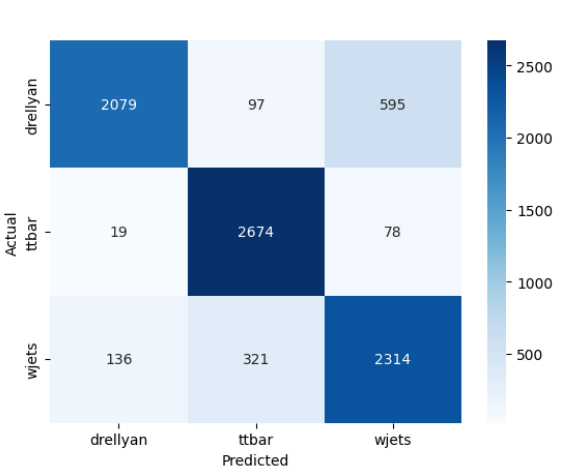



(a) Precisión (b) Valor de pérdida

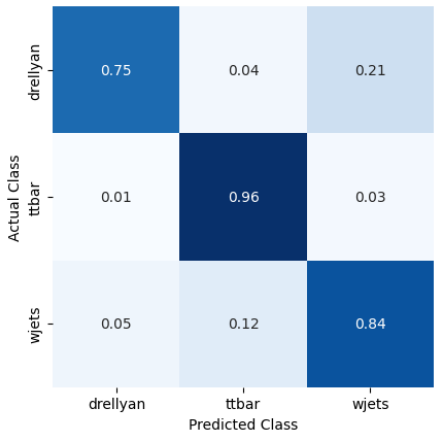
Figura 3.4: Precisión y valor de pérdida del conjunto de validación y entrenamiento en el modelo final entrenado con la base de datos B

Se puede notar en el gráfico de la precisión (a) en las Figuras 3.3 y 3.4 que durante aproximadamente las últimas cinco épocas la precisón evaluada en el conjunto de validación (línea continua) se queda estancada. Pasa de igual forma con el valor de pérdida (b) aunque debido a la escala es más dificil notarlo. Por el

contrario, la precisión evaluada en el conjunto de entrenamiento (puntos) sigue subiendo ligeramente de manera progresiva. Es ésta diferencia la que causa que el modelo deje de entrenarse para prevenir el sobreajuste.

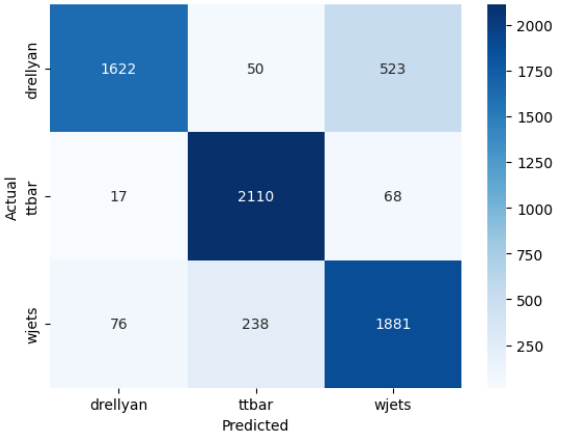



(a) Matriz de confusión no normalizada

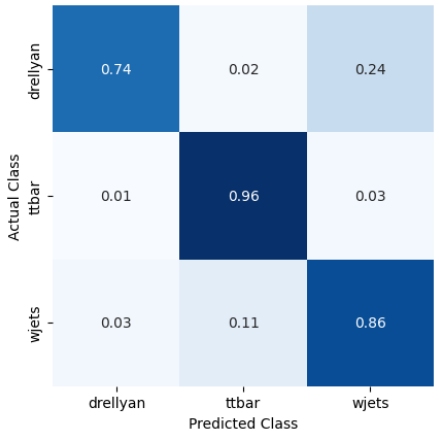



(b) Matriz de confusión normalizada

Figura 3.5: Matrices de confusión del modelo entrenado con la base de datos A




(a) Matriz de confusión no normalizada




(b) Matriz de confusión normalizada

Figura 3.6: Matrices de confusión del modelo entrenado con la base de datos B

Nuevamente se puede observar en las matrices 3.5 y 3.6 que el modelo es ex-

cepcionalmente bueno al momento de clasificar eventos $t\bar{t}$, con un porcentaje de acierto del 96 %. Así mismo, los eventos W + jets fueron clasificados de manera correcta un 84 % de las veces. La categoría en la que peor se desempeñó el modelo fue en Drell-Yan, con un 75 % de aciertos. La mayor confusión ocurre al mal categorizar eventos Drell-Yan y confundirlos con W+jets, esto ocurrió más del 20 % de las veces.

## 3.1. Aplicación del modelo a colisiones reales

Cómo se ha venido discutiendo, una parte importante de usar modelos de clasificación en el ámbito de la física de partículas, es lograr distinguir las resonancias características de determinados procesos. Una muestra de este tipo de resonancias es la encontrada en el proceso Drell-Yan. Previamente se discutió que el resultado final de un evento Drell-Yan puede ser alcanzado también por otros procesos que, sin embargo, no producirán una resonancia similar. El modelo desarrollado deberá, por lo tanto, ser capaz de discriminar las señales y arrojarnos una resonancia limpia pertenenciente al bosón Z.

El modelo que se escogió para hacer las predicciones fue el entrenado con la base de datos completa. A pesar de que tiene una muy ligera peor precisión, el haberlo entrenado con casi el triple de datos lo convierte en un modelo más robusto, esto se ve por ejemplo en su menor valor de pérdida en el Cuadro 3.2. Se corrió el modelo sobre 12722 imágenes pertenecientes a los datos de una colisión real. Si el modelo predecía que una imagen era $t\bar{t}$ o W+jets, se la recategorizaba como "Fakes", si era Drell-Yan permanecía en su misma categoría. Esto con el fin de poder extraer

la información perteneciente al pico característico del bosón Z en los eventos Drell-Yan. Una vez obtenidos los resultados de las predicciones se graficaron las masas invariantes correspondientes de lo que el modelo categorizó como Drell-Yan y como Fakes. En total, de las 12722 imágenes, el modelo categorizó 2266 como Drell-Yan, y 10456 como Fakes.

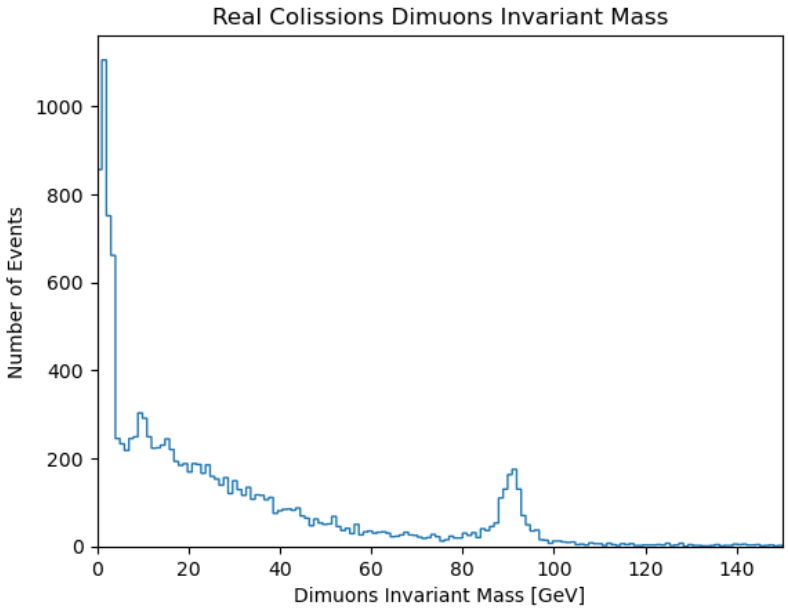



Figura 3.7: Masa invariante de dos muones en una colisión real

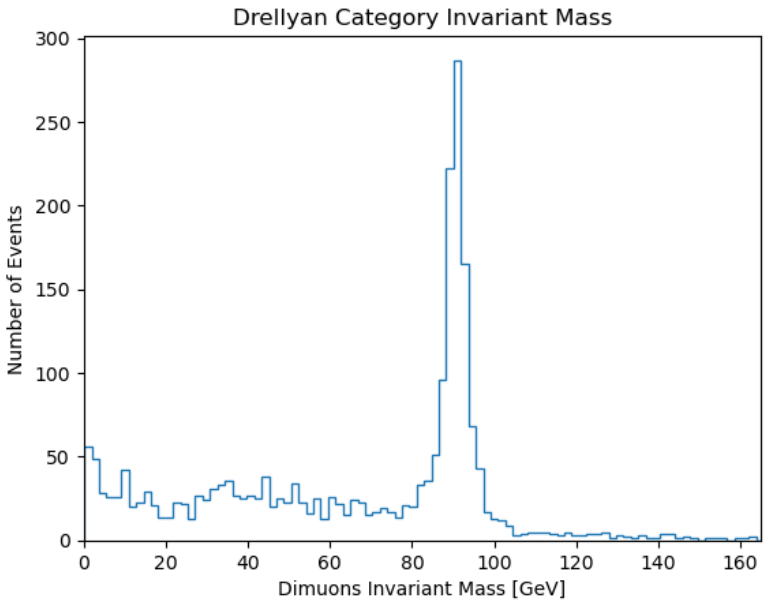



Figura 3.8: Masa invariante de los muones categorizados como Drell-Yan

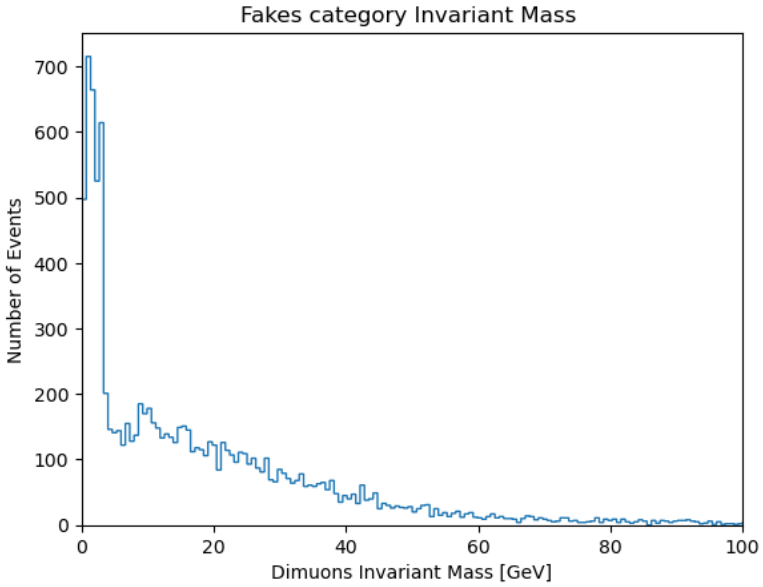



Figura 3.9: Masa invariante de los muones categorizados como Fakes

Como se puede ver en la Figura 3.9 y comparándolo con la Figura 3.7, el pico

que se encontraba alrededor de los 91 GeV desapareció completamente. Así mismo en la Figura 3.8 podemos ver claramente el pico característico que buscábamos. Esto nos quiere decir que el algoritmo fue capaz de separar de una manera convincente este tipo de señales. Finalmente podemos comparar el número de muones de diferente carga presentes en cada una de las predicciones. Para esto tomaremos en cuenta la relación de muones de diferente carga presente en los datasets usados.

| **Categorías** | **Dimuones con diferente carga** [ %] |
|---|---|
| Drell-Yan | 0.8425 |
| $t\bar{t}$ + jets | 0.4710 |
| W + jets | 0.4903 |

Cuadro 3.3: Porcentaje de dimuones con diferente carga en los diferentes procesos

| **Categorías** | **Dimuones con diferente carga** [ %] |
|---|---|
| Drell-Yan | 0.8147 |
| Fakes | 0.5184 |

Cuadro 3.4: Porcentaje de dimuones con diferente carga en las predicciones sobre colisiones reales

Comparando los resultados del Cuadro 3.4 con los del Cuadro 3.3 vemos que la relación es la esperada, con cerca de un 80 % de muones teniendo diferente carga para los eventos Drell-Yan y cerca de un 50 % para los Fakes.

# Capítulo 4

# Conclusiones

Las redes neuronales convolucionales han demostrado a lo largo de los últimos años ser una herramienta muy poderosa para clasificar imágenes pertenecientes a todo tipo de objetos. Esto ha dado paso a que este tipo de modelos poco a poco vayan descubriendo su utilidad en diversas áreas de la Ísica, por ejemplo, en la física de partículas. Existen diversos procesos físicos cuyos estados finales pueden ser muy similares, por lo tanto, difíciles de distinguir unos de otros. Esto puede resultar problemático ya que uno de estos procesos podría ser el causante de una resonancia indicativa de física más allá del modelo estándar o simplemente ser un proceso exótico conocido. En cualquier caso es importante saber distinguir, partiendo del evento final de una colisión, el decaimiento original del mismo.

Usando los datos abiertos del CMS y extrayendo la información necesaria mediante POET se pudo construir una imagen adecuada para un modelo de clasificación basado en redes neuronales convolucionales. Se ha demostrado que se puede usar

la capacidad de los modelos CNN de distinguir patrones para clasificar de manera convincente entre los procesos Drell-Yan, $t\bar{t}$ + jets y W+jets. Se pudo notar también que el modelo es efectivo para detectar el pico característico del proceso Drell-Yan, lo que implica ideas alentadoras en cuanto a usar este tipo de modelos para detectar resonancias pertenecientes a física BSM. La mejora sobre la representación gráfica de las imágenes del modelo de Fernández correspondiente a agregar una pista visual de la carga no fue de mayor ayuda al momento de mejorar la precisión del modelo, sin embargo fue útil para poder hacer una rápida comparación en cuanto a la proporción de muones de distinta carga esperados en los diferentes procesos en las predicciones. El peor desempeño que tuvo el modelo fue categorizando eventos Drell-Yan ya que los confundía con W+jets. Esto resulta ciertamente antintuitivo ya que los dos procesos no comparten similitudes más allá de las estándar en su representación gráfica, no tienen la misma proporción de muones con diferente carga y de manera general los W+jets tienen muchos más eventos con 3 o más jets. Se puede concluir finalmente que los algoritmos y recursos de las redes neuronales convolucionales han resultado tener un gran potencial al ser aplicados en análisis de física de partículas.

Como futuras ideas es interesante buscar nuevas maneras de representar colisiones que puedan resultar en imágenes que se desempeñen mejor al momento de entrenar una red neuronal. Así como realizar análisis similares con diferentes procesos y canales.

# Bibliografía

[1] Izaak Neutelings. CMS coordinate system. Disponible en: https://tikz.net/axis3d_cms/.

[2] Standard Model of Elementary Particle h by MissMJ, Cush. Dominio público. 2019.

[3] CERN. *CMS Open Data Workshop 2022. Disponible en: https://cms-opendata-workshop.github.io/2022-08-01-cms-open-data-workshop/*, 2022.

[4] Izaak Neutelings. Neural networks. Disponible en: https://tikz.net/neural_networks/.

[5] S. Chatrchyan et al. The CMS Experiment at the CERN LHC. *JINST*, 3:S08004, 2008.

[6] Observation of a new boson at a mass of 125 GeV with the CMS experiment at the LHC. *Physics Letters B*, 716(1):30–61, sep 2012.

[7] Ochoa, J. D. CNN-hep-thesis: Undergrad Thesis. using a CNN to classify different HEP processes. GitHub. Retrieved May 5, 2023, from https://github.com/jose8af/cnn-hep-thesis.

[8] CERN Open Data Portal. (n.d.). https://opendata.cern.ch/.

[9] Cms-Sw. (n.d.). GitHub - cms-sw/cmssw: CMS Offline Software. GitHub. https://github.com/cms-sw/cmssw.

[10] Cms-Opendata-Analyses. (n.d.). GitHub - cms-opendata-analyses/PhysObjectExtractorTool: This repository has working code examples (snippets) on how to access different physics objects in the context of CMSSW software. GitHub. https://github.com/cms-opendata-analyses/PhysObjectExtractorTool.

[11] R. L. Workman et al. Review of Particle Physics. *PTEP*, 2022:083C01, 2022.

[12] Celia Fernández Madrazo, Ignacio Heredia Cacha, Lara Lloret Iglesias, and Jesús Marco de Lucas. Application of a Convolutional Neural Network for image classification for the analysis of collisions in High Energy Physics. *EPJ Web Conf.*, 214:06017, 2019.

[13] Tensorflow documentation, [online]. disponible en: https://www.tensorflow.org.

[14] Keras documentation, [online]. disponible en: https://keras.io.

[15] I. goodfellow, y. bengio and a. courville, deep learning. mit press, 2016, http://www.deeplearningbook.org.

[16] Goodfellow, Ian; Bengio, Yoshua; Courville, Aaron (2016). "6.2.2.3 Softmax Units for Multinoulli Output Distributions". Deep Learning. MIT Press. pp. 180–184. ISBN 978-0-26203561-3.

[17] CMS collaboration (2021). DoubleMuon primary dataset in MINIAOD format from RunD of 2015 (/DoubleMuon/Run2015D-16Dec2015-v1/MINIAOD). CERN Open Data Portal. DOI:10.7483/OPENDATA.CMS.H3TX.ZJZX.

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

[19] Imagenet (n.d). disponible en: http://www.image-net.org.

[20] Min Lin, Qiang Chen, and Shuicheng Yan. Network in network, 2014.

[21] Diederik kingma and jimmy ba. adam: A method for stochastic optimization, 2014.

[22] Kevin P. Murphy. *Probabilistic Machine Learning: An introduction.* MIT Press, 2022.

[23] Google colab. Disponible en: https://colab.research.google.com/ (Accedido el: 11 May 2023).