Clasificación de objetos cosmológicos usando Redes Neuronales Convolucionales

Enrique A. Galceran García

MÁSTER EN ASTROFÍSICA. FACULTAD DE CC. FÍSICAS
UNIVERSIDAD COMPLUTENSE DE MADRID

Septiembre 2019

Directores:

Dr. Ignacio Sevilla Noarbe
Dr. Miguel Cárdenas Montes

Resumen en castellano

En los últimos 20 años, la tecnología de detectores y procesamiento de datos ha permitido que hoy en día, los astrónomos dispongan de una inmensa cantidad de datos, tanto de objetos particulares, como de amplias áreas del cielo. Durante los primeros años de esta revolución, el post-procesado de los datos, como en el caso de la clasificación de objetos, era realizado manualmente por los científicos.

Hoy en día, los instrumentos modernos nos permiten obtener fotometría de miles de objetos cada noche en todo el mundo. Para poder analizar toda esta información de manera eficiente, hay que crear sistemas automatizados de clasificación. El objetivo de este trabajo consiste en desarrollar un sistema para poder analizar la enorme cantidad de datos generada por los nuevos sistemas automatizados. Para ello, utilizaremos *machine learning* (aprendizaje automático) para analizar los espectros que tomamos usando fotometría con filtros estrechos y poder separar entre galaxias, estrellas y cuásares de una forma rápida, eficaz y fiable.

Palabras clave

Clasificación, espectros, morfología, redes neuronales, convolución, CNN, DBSCAN, PCA

Abstract

During the last 20 years, the technology on detectors and data processing has allowed astronomers to have nowadays an immense amount of data, both from individual objects, as from large areas of the sky. During the first years of this revolution, the post-processing of the data, as in the case of the classification of objects, was done manually by the scientists.

Today, modern instruments allow us to obtain photometry of thousands of objects every night throughout the world. In order to analyze all this information efficiently, an automated classification systems has to be created. The objective of this Master's thesis is to develop a system to analyze the huge amount of data generated by the new automated systems. To do this, we will use machine learning to analyze the spectra we take using photometry with narrow filters and to be able to separate between galaxies, stars and quasars in a fast, efficient and reliable way.

Índice general

Agradecimientos

Quiero agradecer a Ignacio Sevilla Noarbe y a Miguel Cárdenas Montes por la oportunidad que me han dado para poder realizar este trabajo fin de máster con ellos y por siempre haber estado disponibles para responder mis dudas y preguntas. Sin sus indicaciones, guiado, consejos y ayudas en estos últimos meses (incluso en fines de semana) no habría sido posible realizar este trabajo.

También quiero agradecer a los estudiantes de doctorado que se encuentran en el departamento de Altas Energías en el CIEMAT por haberme acogido y dado consejos durante la realización de este trabajo fin de máster.

Adicionalmente quiero agradecer al CIEMAT por haberme permitido usar sus instalaciones y equipamiento.

1. Introducción

En los últimos años hemos recibido los volcados de datos de telescopios como Gaia [Gaia Collaboration[8]] o vamos a recibirlos en el futuro como es el caso de TESS [Ricker[1516]], que nos han proporcionado una inmensa cantidad de información que es necesario analizar. Sin embargo, el análisis de toda esa información es inviable de modo tradicional. Esto empuja a desarrollar sistemas que sean capaces de procesar toda esa información de forma automática, autónoma y sin necesidad de supervisión constante.

Uno de los primeros análisis necesarios es catalogar cuales son los objetos que se están estudiando. En el óptico uno de los sistemas que se usan ahora mismo consiste en comparar la extensividad de los objetos celestes y catalogar así entre estrellas y galaxias [Bertin and Arnouts[3]]. Este método es bueno para diferenciar entre objetos extensos y puntuales, pero hay otros tipos de objetos cosmológicos con los que este método no es suficiente.

Además, cuando la relación señal/ruido (S/N) se vuelve muy baja, estas medidas se vuelven muy imprecisas y estos métodos poco fiables. Simultáneamente, si aumentamos el tiempo de exposición para corregir estos problemas, nos encontramos con que la función de dispersión de punto (Point Spread Function) podrá además recibir señales de estrellas más lejanas y menos brillantes que dificulten la categorización. Las estrellas dobles también suponen una importante fuente de confusión.

Actualmente existen métodos que utilizan información pseudo-espectral para clasificar objetos en galaxias, estrellas y cuásares. Estos métodos utilizan información de filtros de banda ancha en el infrarrojo y el ultravioleta para comparar las magnitudes entre diferentes de estos filtros. [Baldry et al.[2]]

En L. Cabayol et al. [Cabayol et al.[4]] se presenta un método que soluciona los problemas de la clasificación por morfología. En este trabajo se implementa una red neuronal convolucional (CNN) basada en la fotometría de bandas estrechas en el visible, frente a otras aproximaciones que se basan en la fotometría espacial. Al usar 'espectros de baja resolución', se abre la posibilidad de encontrar una mayor variedad de objetos, no únicamente estrellas y galaxias.

En este trabajo fin de máster continuaremos el trabajo realizado en L. Cabayol et al. También se mantiene el catálogo PAU Survey (Physics of the Accelerated Universe Survey) [Castander et al.[5], Martí et al.[14]] como datos de reference, debido a la gran cantidad de objetos que ya han sido catalogados allí, y para poder realizar comparativas con el trabajo previo. Para etiquetar los diferentes objetos, usaremos la alta resolución espacial que tiene el telescopio espacial Hubble (0.05 arcsec), pudiendo asegurarnos de la fiabilidad de esa medida. Adicionalmente, una ventaja de usar machine learning reside en que, una vez entrenada es capaz de categorizar nuevos objetos de forma autónoma.

Existen múltiples métodos que usan la extensión de los objetos en las imágenes fotométricas en redes neuronales y CNNs para la clasificación de objetos extensos y puntuales [Kim and Brunner[10]], pero ninguno aprovecha los espectros existentes en PAU. El caso ideal

consistiría en generar una CNN doble que clasifique usando los espectros 1D y las imágenes 2D. En este trabajo sea realizará una primera aproximación que utilizará los espectros así como la concentración de luz del objeto.

Así pues, aquí ampliaremos el sistema dicotómico (galaxias/estrellas) del trabajo previo incluyendo objetos cuasi estelares (quasars/cuásares). La dificultad de esta inclusión estriba en la escasa cantidad de objetos de esta clase disponibles. Además, al añadir las propiedades morfológicas a la red, mejoramos la red neuronal, obteniendo con ello una precisión mayor que con los otros dos métodos por separado.

2. Datos

En este trabajo se usarán los datos obtenidos del PAU Survey [Castander et al.[5], Martí et al.[14]] del campo COSMOS para la obtención de los espectros a analizar. Dichos espectros estarán formados por imágenes fotométricas de 40 bandas estrechas de 10nm de ancho de banda. Para la obtención de las etiquetas, se usará el catálogo morfológico generado a partir de imágenes del telescopio espacial Hubble (HST) con la Advanced Camera for Surveys (ACS) [Leauthaud et al.[12]]. Estas imágenes son muy profundas y de muy alta calidad, al no ser afectadas por la atmósfera. Para el reconocimiento de los cuásares, se compararán con los diferentes objetos del Million Quasars Catalog (MILLIQUAS) [Flesch[7]].

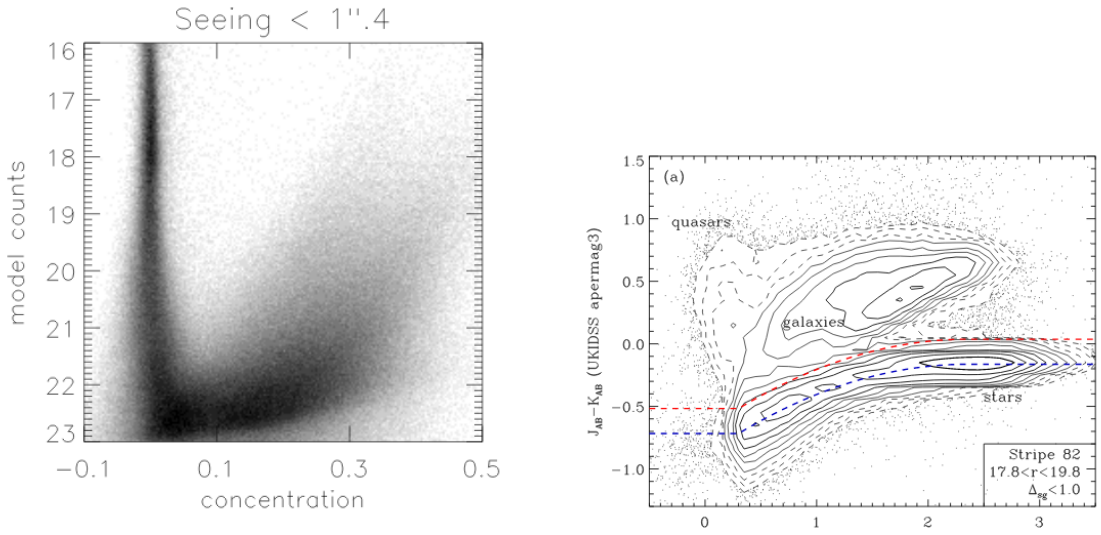
3. Herramientas usadas en este trabajo para la clasificación de objetos cosmológicos

3.1. Selección por cortes

El uso de cortes sencillos en variables morfológicas es la primera aproximación que puede hacerse para resolver este problema. Originalmente, la medida de concentración de luz frente a magnitud constituía un método razonable para la separación de estrellas y galaxias. Durante los últimos años, el uso de redes neuronales para catalogar imágenes fotométricas han ido aumentando hasta que hoy en día sean igual de frecuentes.

Las estrellas tienen un perfil de luz que se corresponde a la función de dispersión de punto (*Point Spread Function*/PSF) de la atmosfera más el instrumento, mientras que las galaxias típicamente tendrán un perfil exponencial de algún tipo, convolucionada con dicha PSF. La comprobación de si un objeto en cuestión se ajusta más a un modelo morfológico u otro suele ser un método adecuado de clasificación de acuerdo a este criterio (que empieza a tener menor eficacia a medida que observemos galaxias más y más lejanas, que se aproximen por tanto a la PSF a su vez).

El ajuste de la morfología de un objeto extenso y uno puntual con estos dos tipos de modelos para una estrella proporciona un valor similar para estrellas (la magnitud usando la PSF del sistema en dicho punto es similar a la magnitud de un modelo exponencial ajustado a la misma estrella). Para una galaxia en cambio, proporciona valores dispares, ya que la PSF no ajusta adecuadamente. Así pues, con la resta de ambos parámetros, podemos generar un valor de la 'concentración de la luz'. La representación de ese parámetro junto a la magnitud nos permite generar secciones en la que se separarán las estrellas de las galaxias (Fig. 3.1(a)) [Scranton et al.[17]].



(a) Clasificación entre objetos extensos y puntuales usando la magnitud del objeto frente a la concentración.

(b) Separación por cortes para clasificar entre galaxias, estrellas y cuásares. Para poder realizar esta medición se precisa de valores de colores en el infrarrojo.

Figura 3.1: *Diferentes métodos de clasificación de imágenes mediante cortes. Figura (a) procedente de Figura 1 de [Scranton et al.[17]], figura (b) procedente de Figura 6.a de [Baldry et al.[2]]*

Si se tienen los valores de magnitudes para los objetos en el infrarrojo o en el ultravioleta, se puede realizar una separación por colores. Este método nos permite separar una mayor cantidad de objetos usando cortes, pero son imágenes científicas más difíciles de obtener con una alta relación S/N debido al tiempo de exposición necesario además de la dificultad de obtener dichas medidas con instrumentos terrestres (Fig. 3.1(b)) [Baldry et al.[2]].

Uno de los problemas que tiene este método se hace aparente cuando observamos dos fuentes puntuales muy cercanos. En esos casos, una estrella doble se reconoce como un objeto extenso. Estas estrellas dobles pueden surgir al resolver un sistema binario o porque haya una estrella lejana y brillante que en la imagen fotométrica se acerque a la estella cercana.

Otro problema que podemos tener es la resolución espacial de los telescopios terrestres. Si la resolución es limitada, una galaxia pequeña puede aparentar una fuente puntual cerca del límite de difracción. Es por eso que las etiquetas que usamos para clasificar nuestros cuerpos

se han obtenido del telescopio espacial Hubble. Como el HST es un telescopio espacial, no sufre los efectos del seeing atmosférico y nos permite obtener una precisión mucho mayor para diferenciar entre los diferentes objetos.

3.2. Machine Learning

El término Aprendizaje Automático (*Machine Learning*) describe un conjunto de algoritmos capaces de aprender modelos basados en datos. Las dos áreas dentro del Aprendizaje Automático más relevantes en este trabajo son los denominados *aprendizaje supervisado* y *aprendizaje no supervisado*.

Los algoritmos denominados *aprendizaje supervisado* entrenan sus redes usando pares de ejemplos con sus etiquetas. Estas etiquetas pueden ser de tipo continuas para hacer regresión o de tipo discretas cuando se está realizando clasificación. En nuestro caso usaremos etiquetas discretas, separando por galaxias, estrellas y cuásares. Por otro lado, los algoritmos de *aprendizaje no supervisados* se entrenan con ejemplos sin etiquetas. Estos algoritmos clasifican según la similitud entre los diferentes objetos disponibles.

De entre estos métodos uno de los más conocidos y usados son las redes neuronales, siendo aplicado en una infinidad de lugares hoy en día, desde identificar objetos en una imagen a calcular pólizas de seguro.

Las redes neuronales se llaman así porque simulan el comportamiento de las neuronas dentro del cerebro. Una neurona individual ejerce de puerta lógica sencilla que, al juntarse con otras neuronas, pueden formar una estructura compleja que es lo que se llama Red Neuronal. Los parámetros que modulan las conexiones entre neuronas se pueden ir modificando para ajustar los resultados finales. Si el resultado de la red neuronal es igual o parecido al de los datos de entrenamiento, se modificarán dichos parámetros para reforzar las conexiones que se han activado; mientras que, si el resultado es erróneo se modificarán los parámetros para corregir ese error.

En astronomía, uno de los métodos de clasificación de objetos extensos y puntuales más usados es el incorporado en el software SExtractor [Bertin and Arnouts[3]]. Este método utiliza redes neuronales usando la información de los píxeles para diferenciar entre los diferentes tamaños y formas de los objetos.

En nuestro caso vamos a usar un tipo específico de redes neuronales llamadas redes neuronales Convolucionales (CNN/*Convolutional Neural Network* en inglés). Estas redes tienen la ventaja de que nos permiten realizar un análisis de patrones que se forman en los datos.

3.2.1. Funcionamiento de las redes neuronales

Fundamentalmente, el proceso de generar redes neuronales es la búsqueda de una aproximación de una función con muchas variables de entrada a partir de una gran cantidad de ecuaciones más simples.

3.2.1.1. Estructura

Una neurona aislada es una ecuación consistente en un sumatorio (Eq. 3.1). A cada parámetro de entrada se le da un peso (también llamado peso sináptico) que cuantifica la

importancia que tiene ese valor. Posteriormente, se multiplican los valores de entrada por los pesos, se suman todas las entradas y se evalúa ese resultado con una función de activación. Estas funciones son en su gran mayoría no lineales. Debido a esto, las redes neuronales son capaces de generar estas aproximaciones de las funciones buscadas.

$$x_k^j = y_k^{j-1} = f\left(\sum_i^n x_i^{j-1} \cdot \omega_{i,k}^{j,j-1}\right) \tag{3.1}$$

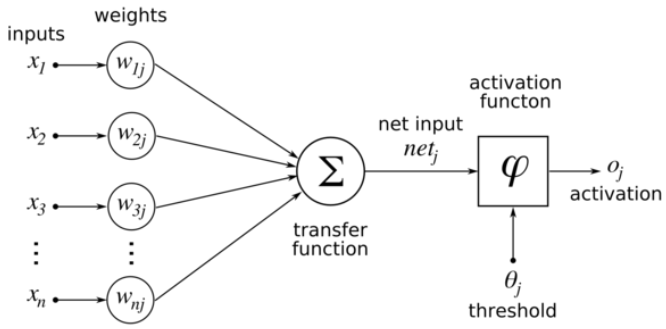


Figura 3.2: *Estructura de una única Neurona*

Las funciones de activación pueden ser sencillas, como la función de paso Heaviside, o más complicadas como la función logística sigmoide o la tangente hiperbólica. En nuestra red usaremos la variante *Leaky ReLU* (Eq. 3.2b) de la función *ReLU* (Eq. 3.2a) (*Rectified Linear Unit*). Las funciones de activación regulan la cantidad de información que se transmite de neurona a neurona. Si estamos trabajando con una red muy grande, nos interesará usar una función como *ReLU*, pues los valores negativos pasarán a ser 0 y realizar muchas operaciones con matrices dispersas acelera el proceso de evaluación, así como el método de aprendizaje que se esté usando.

$$ReLU(x) = \begin{cases} 0, & x < 0 \\ x, & x \geq 0 \end{cases} \tag{3.2a}$$

$$Leaky\ ReLU(x) = \begin{cases} x \cdot \alpha, & x < 0 \\ x, & x \geq 0 \end{cases}, \alpha \lesssim 0{,}1 \tag{3.2b}$$

La combinación de múltiples neuronas combinadas en paralelo nos proporciona una capa. La unión de múltiples de estas capas en serie resulta en una red neuronal:

- **Capa de entrada:** Son los valores que hemos introducido de entrada y queremos evaluar en cada caso. Transmiten la información a la primera capa oculta.
- **Capas ocultas:** Estas neuronas reciben como entrada las neuronas de las capas anteriores. Los valores de salida que se obtienen para cada una de estas neuronas se utilizarán posteriormente como entrada para las neuronas de la siguiente capa. Se llaman capas ocultas, porque el usuario no interacciona directamente con ellas.

- **Capas de salida:** Esta capa puede tener una o más neuronas. La entrada de esta capa son los valores de salida de las neuronas de la última capa oculta y su salida es la que verá el usuario.

3.2.1.2. Aprendizaje

Las redes neuronales tienen la capacidad de ajustar sus pesos de manera independiente para obtener el mejor resultado posible. Los datos deben estar separados entre los datos de entrenamiento y los datos de validación. Esta separación se realiza para que haya un conjunto de datos con los que la red no entrene y poder usarlos posteriormente para comprobar la eficacia de dicha red.

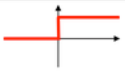
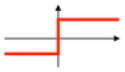
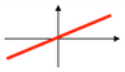
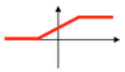
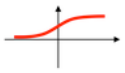
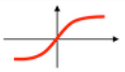
Activation function	Equation	Example	1D Graph
Unit step (Heaviside)	$\phi(z) = \begin{cases} 0, & z < 0, \\ 0.5, & z = 0, \\ 1, & z > 0, \end{cases}$	Perceptron variant	
Sign (Signum)	$\phi(z) = \begin{cases} -1, & z < 0, \\ 0, & z = 0, \\ 1, & z > 0, \end{cases}$	Perceptron variant	
Linear	$\phi(z) = z$	Adaline, linear regression	
Piece-wise linear	$\phi(z) = \begin{cases} 1, & z \geq \frac{1}{2}, \\ z + \frac{1}{2}, & -\frac{1}{2} < z < \frac{1}{2}, \\ 0, & z \leq -\frac{1}{2}, \end{cases}$	Support vector machine	
Logistic (sigmoid)	$\phi(z) = \frac{1}{1 + e^{-z}}$	Logistic regression, Multi-layer NN	
Hyperbolic tangent	$\phi(z) = \frac{e^{z} - e^{-z}}{e^{z} + e^{-z}}$	Multi-layer NN	

Figura 3.3: *Una muestra de diferentes funciones de activación que se pueden usar*

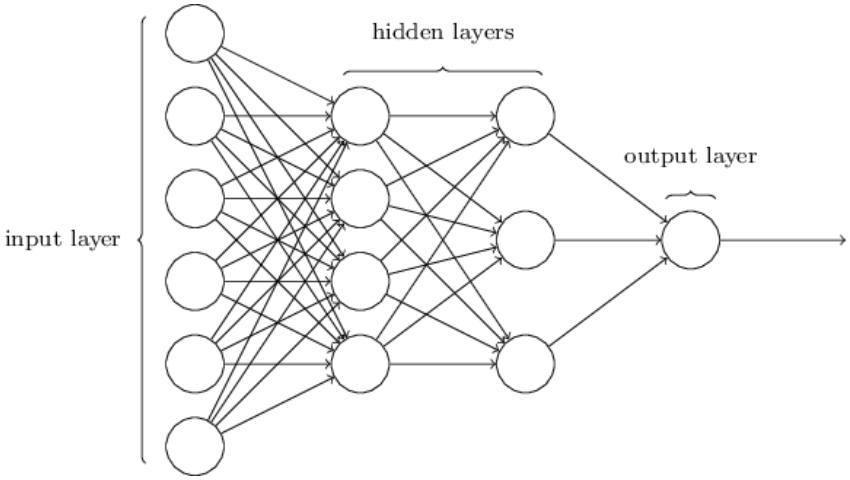


Figura 3.4: *Estructura de una red neuronal con 6 entradas, dos capas ocultas y una salida.*

Los datos de entrenamiento los introduciremos en la red y evaluaremos la salida que obtenemos. Debido a que inicialmente los pesos en cada conexión sináptica entre las neuronas estarán generado al azar (generalmente con una distribución), las primeras predicciones serán erróneas. Podemos evaluar cuán errónea es dicha predicción usando la *función coste* (o también llamada *función pérdida*) (Eq. 3.3).

$$C(\omega, b) = \frac{1}{2n} \sum_{x} \|\hat{y}(x) - a\|^2 \tag{3.3}$$

$$H(\omega, b) = \frac{1}{N} \sum_{i=1}^{N} [y_i \log(\hat{y}_i(x) + (1 - y_i) \log(1 - \hat{y}_i(x))] \tag{3.4}$$

Esta función evalúa las salidas de cada neurona de salida (y) y le resta el valor real (a). Aquí de nuevo hay varias versiones de función de coste que pueden ayudar a acelerar el aprendizaje. En este trabajo hemos utilizado la función de coste de entropía cruzada de la teoría de la información (Eq. 3.4).

La ventaja de usar este proceso es que podemos obtener, derivando en cadena, la influencia que ha tenido cada uno de los pesos de la red en esta función coste. Así pues, en cada época de entrenamiento (cada ciclo de procesamiento de todos los datos de entrenamiento) podemos modificar ligeramente dichos pesos con el objetivo de minimizar la función coste. Este proceso de corregir los pesos se llama *Backpropagation* (retropropagación) (Eq. 3.5).

$$\omega^l_{i,k} \to \omega^l_{i,k}{}' = \omega^l_{i,k} - \frac{\eta}{m} \sum_{i}^{n} \frac{\partial C_{X_i}}{\partial \omega^l_{i,k}} \tag{3.5}$$

donde ω son los pesos, η la velocidad de entrenamiento, l el número de la capa de la neurona asociada al peso, m el número de neuronas en la capa $l-1$, n el número de neuronas en la capa de entrada y X_j cada una de dichas entradas. El paso de la información de capa a capa se puede escribir como un producto vectorial, permitiendo calcular la diferencia para cada peso capa a capa (Eqs. 3.6).

$$\begin{aligned} f\left(\overrightarrow{X^l} \cdot \overrightarrow{\omega^l}\right) &= \overrightarrow{X^{l+1}} \\ f\left(...f\left(f\left(\overrightarrow{X^1} \cdot \overrightarrow{\omega^1}\right) \cdot \overrightarrow{\omega^2}\right) \cdot ... \cdot \overrightarrow{\omega^{salida}}\right) &= \vec{Y} \end{aligned} \tag{3.6}$$

Esto nos ayudará entre otras cosas a mejorar la eficacia del aprendizaje. A la hora de aprender, en lugar de usar un solo caso para aprender a cada vez, se pueden usar múltiples casos simultáneamente. Este grupo de entrenamiento se le llama *training batch* (lotes de entrenamientos). El aprendizaje puede ser optimizado mediante el ajuste de ciertos parámetros como la velocidad de entrenamiento.

3.2.1.3. Validación de resultados

Para comprobar la eficacia de nuestra red, realizaremos varias comprobaciones. Uno de los puntos débiles que tienen las redes neuronales es el sobreajuste (*overfitting*). Este

problema aparece cuando el modelo entrenado no se generaliza bien a nuevos datos debido a la especificidad de la muestra de entrenamiento.

Las redes neuronales se ajustan a los datos dados. Si ha entrenado correctamente, la red se ajusta correctamente al modelo que se busca analizar. En el caso de sobreajuste, la red está produciendo una clasificación basada en la muestra que se haya dado, en lugar de la modelo que los datos representan.

Para verificar si ocurre esto, podemos usar los datos de validación que separamos al principio. Estos datos, como la red no ha entrenado con ellos, se pueden usar para comprobar la efectividad para el caso general.

Si representamos la función coste que obtenemos con los datos de entrenamiento y validación podemos ver cómo va disminuyendo a cada época de entrenamiento. Esto va ocurriendo hasta que lleguemos a los pesos óptimos para cada sinapsis. Si seguimos entrenando, debido al sobreajuste la función coste del entrenamiento va a seguir reduciéndose, pero para los datos de validación va a volver a ir aumentando.

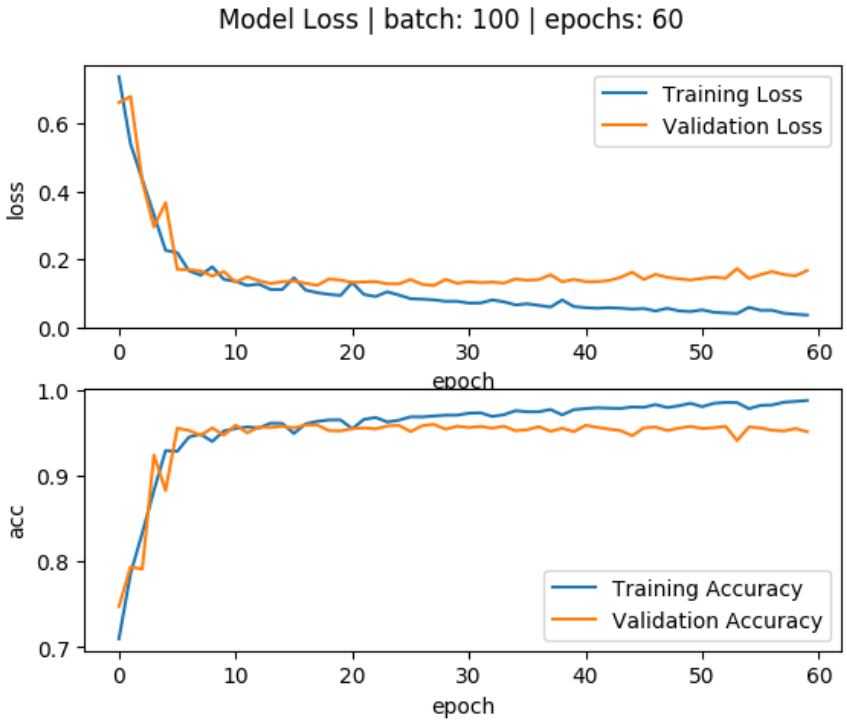


Figura 3.5: *Comparación de la función coste y la precisión en la predicción para la red neuronal que usaremos en la sección 4. Se empieza a ver sobreajuste a partir de la época 15. Para este caso, se usan los pesos de la red en dicha época.*

3.2.2. Redes neuronales Convolucionales

Las redes neuronales convolucionales (CNN/*Convolutional Neural Networks* en inglés) son un tipo de redes neuronales que se especializan en encontrar patrones locales. A la hora

de asignar las variables de entrada a las neuronas de la primera capa de una red neuronal convencional, no influye la posición de cada una de dichas variables. En nuestro caso, si asignamos las bandas de rojo a azul vamos a obtener el mismo resultado que si las ordenamos de azul a rojo. Una de la diferencia que tienen las redes neuronales convolucionales con las convencionales es que las asignamos de una forma desordenada, dejará de ser efectiva. En el caso de las redes neuronales convolucionales, las neuronas se analizan usando unos filtros, que comparan los valores de cada neurona con sus adyacentes. Esto permite reconocer patrones en las entradas de datos, como en el caso de reconocimiento de fotos o en este trabajo [Goodfellow et al.[9]].

En las CNN se generan una serie de capas al principio de la red que generan unos filtros (*kernels* en inglés) para buscar patrones que condensarán la información. Estos filtros van pasando por cada valor de la entrada y evaluando cada conjunto de puntos.

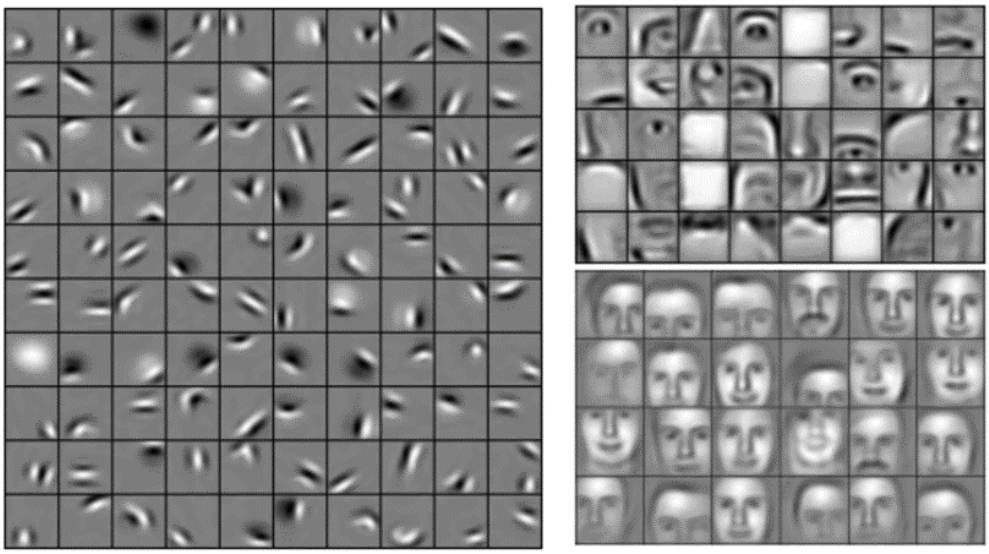


Figura 3.6: *Filtros intermedios para una CNN que reconoce caras. Figura adaptada de [Lee et al.[13]]*

Para mostrar cómo funciona una CNN, un ejemplo gráfico puede ser el reconocimiento de caras en una imagen (Fig. 3.6). En la primera capa convolucional (izquierda), en los filtros que se han generado solamente se reconocen pequeños fragmentos y patrones frecuentes. En la segunda capa (derecha arriba), los filtros buscan combinaciones de los patrones detectados en la primera capa que formen diferentes partes de una cara. Finalmente, la tercera capa convolucional (derecha abajo) busca diferentes patrones de caras que pueden encontrarse en las imagenes de entrenamiento.

Cada uno de los filtros se especializa en encontrar patrones específicos una vez entrenados. Una CNN generalmente contiene múltiples capas convolucionales, intercalándolas con *pooling layers*. En este trabajo usamos las capas *max pooling layers*, que evalúan una matriz por todos los valores de la capa anterior quedándose con el valor máximo de aquellos que caen en su casilla (Fig. 3.7); reduciendo el tamaño de las capas para consolidar la información.

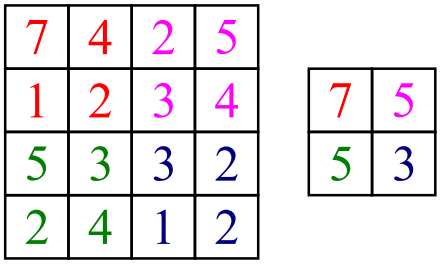


Figura 3.7: *Entrada y salida de una matriz max pooling 2D de tamaño 2x2.*

Estas capas también son no lineales. Después de pasar por múltiples capas de convolución, estas salidas (llamadas características o atributos, *features*) se vuelven a introducir en una red neuronal.

Los filtros que se generan en cada paso los decide la red neuronal a medida que va entrenando todas las capas simultáneamente. Es frecuente que estos filtros aparenten no tener un objetivo claro a la vista de un humano. Si un filtro encuentra el patrón específico para el que ha entrenado, la salida para ese filtro será muy alta. Existen CNNs con grandes cantidades de datos de entrenamiento, como la *Deep Neural Network de Google*, en las que se puede apreciar qué están buscando algunos de los filtros: existencia de pelo o plumas, ojos, diferentes tipos de orejas, ruedas, caras humanas, etc.

En este trabajo, estamos realizando la clasificación de espectros 1D. Es por eso que los filtros generados en la CNN que usaremos en este trabajo debe ser también 1D. En la figura (Fig. 3.8) se representan los atributos aprendidos tras la primera capa convolucional de la CNN usada en este trabajo. En vertical se representan los 32 filtros que se generan uno encima del otro para facilitar la representación. Se puede ver cómo se activan de forma diferente para cada tipo de objetos cosmológicos.

La ventaja de utilizar CNNs reside en que además de generar una información más compacta, útil y eficiente para el programa, nos permite obtener los atributos features resultantes de las capas CNN, y especialmente de la última de ellas, que podremos usar más adelante para realizar otros análisis (como PCA o DBSCAN, que veremos en las secciones 5.1 y 5.2).

4. Desarrollo de una nueva CNN que clasifique galaxias, estrellas y cuásares

El sistema desarrollado en Cabayol et al.[4] era capaz de separar entre estrellas y galaxias, a partir de sus características fotométricas con PAUCam. El conjunto de datos usado en ese trabajo estaba compuesto por 6 mil objetos de clase estrella y 24 mil de clase galaxia, identificados según las observaciones de Leauthaud et al.[12] con el instrumento ACS del Hubble Space Telescope. El objetivo de este trabajo consiste en generar una red que sea capaz de separar adicionalmente entre cuásares, utilizando la identificación provista por la

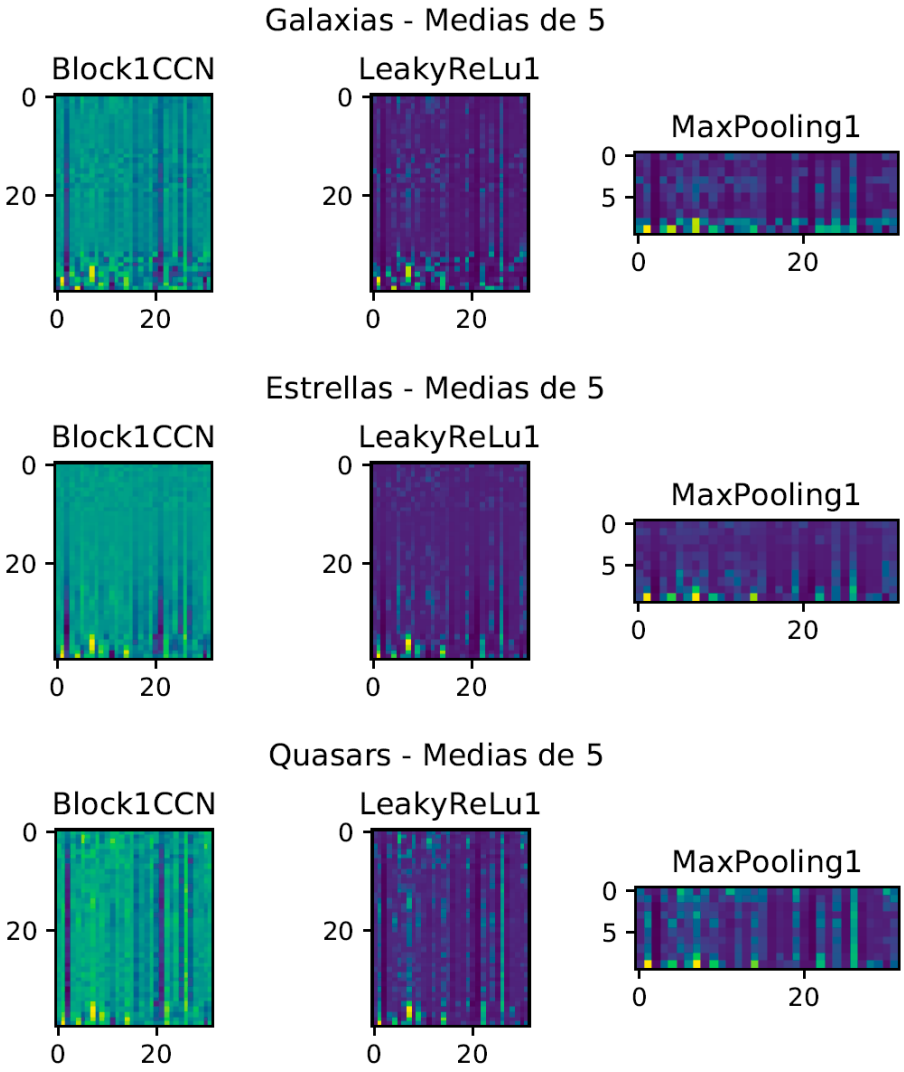


Figura 3.8: *Primeros filtros de la CNN una vez entrenada.*

base de datos Milliquas Catalog[6]. La dificultad añadida a la inclusión de una nueva clase de objetos (además del tratamiento multiclase) es el fuerte desbalanceo en contra de los cuásares, al disponer de solo 100 cuásares.

4.1. CNN con clases desbalanceadas

Una solución inicial al problema consiste en ignorar el desbalanceo y simplemente ampliar las posibles salidas de la red para que acepte una separación categórica. En Cabayol et al.[4] se separaba únicamente entre dos clases, luego una salida Verdadero/Falso era suficiente. En este trabajo, al estar separando entre tres clases diferentes, una salida booleana no es capaz de mostrar las diferentes combinaciones, haciendo falta una salida categórica. Esto es, que de tres posibles resultados, los que corresponden con la confianza de la red de que

formen parte de cada uno de las tres clases (siendo la suma 100 %). Para poder comprobar la confianza de la red, separaremos los datos en entrenamiento y validación (siendo los datos de validación el 30 % de cada clase).

Para calcular la eficacia de la red, utilizaremos las matrices de confusión. Dichas matrices comparan la precisión para la predicción de cada clase. Las salidas categóricas de la red nos darán el valor de confianza de que un objeto pertenezca a una clase u otra. A la hora de decantarnos por cual es la salida para un objeto dado, nos quedaremos con la clase con la mayor probabilidad.

Una vez tengamos la red entrenada, el porcentaje de acierto es del 96 %. Sin embargo, ese número no es representativo de la eficacia real de la red. Al usar las matrices de confusión, el porcentaje de éxito resultante cambia drásticamente, no siendo capaz de clasificar los cuásares (cuadro 4.1).

Clase real\Predicción	Galaxias	Estrellas	Cuásares	% Éxito
Galaxias	9774	227	0	97.6
Estrellas	188	915	0	83.0
Cuásares	23	6	0	0

Cuadro 4.1: *Matriz de confusión de los datos de validación para el caso desbalanceado.*

El aprendizaje de las redes neuronales funciona por pasos, donde a cada iteración va cambiando los valores de los parámetros para que cada vez clasifique mejor, luego a la hora de entrenar sólo 1 de cada 300 casos contiene un cuásar. Siguiendo esa baja cantidad de este tipo de ejemplos, es posible que termine por no considerar los cuásares y concluye que no existen los cuásares. Esta solución que ha encontrado la red efectivamente significa que de cada 300 veces sólo falla 1 de los casos, pero estamos buscando justo ese caso especial.

Aumentar el tiempo de entrenamiento podría eventualmente corregir el problema, pero de lograrlo, sería a cambio de un sobreajuste muy severo. Esta es la razón de usar los datos de validación. Debemos pues, encontrar una solución alternativa al problema.

4.2. Cómo corregir datos desbalanceados

La forma más directa de corregir datos desbalanceados consiste en ampliar los ejemplos de las clases minoritarias. Sin embargo, la cantidad de cuásares que se han detectado es muy baja, lo cual reduce la disponibilidad de espectros para poder balancear los datos de entrenamiento. Una posible solución consiste en usar dos técnicas para aumentar la cantidad de cuásares en nuestra muestra generando datos nuevos o repitiéndolos: *Synthetic Minority Over-sampling Technique* (SMOTE) [Last et al.[11]] y *Random OverSampling* (ROS).

4.2.1. Synthetic minority Over-sampling Technique (SMOTE)

El algoritmo SMOTE busca entre los diferentes datos de la clase que se desea aumentar un par de puntos que sean vecinos próximos y genera un objeto nuevo que se encuentre entre esos dos en el espacio de fases. Este método es muy útil cuando se rebalancean datos para entrenamiento de redes neuronales, pues nos interesa que los valores nuevos que se han generado sean valores diferentes a los que ya estemos observando en el sistema.

4.2.2. Random OverSampling (ROS)

Otro método para corregir los datos desbalanceados consiste en repetir aquellos objetos de la clase minoritaria para igualar la frecuencia de cada clase. Este método va generando puntos nuevos repitiéndolos al azar de entre los originales para evitar un posible sesgo en la medida.

La ventaja de ROS con respecto a SMOTE reside en los espectros que se están generando. SMOTE genera espectros artificiales a partir de los espectros de cuásares que tenemos disponibles. Estos objetos nuevos de SMOTE tienen la desventaja de ser objetos que no han sido observados, arriesgándonos a que no tengan sentido físico; mientras que con ROS estamos duplicando objetos que sí han sido observados. Si entrenásemos con los espectros generados por SMOTE, existe la posibilidad de que cuando se vaya a evaluar un cuásar real con el que no había entrenado previamente no sea capaz de reconocer dicho objeto como un cuásar.

4.2.3. Producto de rebalancear los datos

La aplicación de ambos algoritmos sobre los datos iniciales resulta en una CNN que es capaz de obtener un ajuste para las tres clases (cuadro 4.2). La utilización de estos métodos de rebalanceo reduce un 1-2 % los resultados con respecto a la red inicial desbalanceada para las galaxias y estrellas, a cambio de poder identificar los casos particulares de los cuásares con frecuencia 1/300.

Adicionalmente, la utilización de ROS respecto a SMOTE no cambia los resultados de forma apreciable, por lo que seguiremos usando ROS de ahora en adelante con la intención de mantener el significado físico de los nuevos espectros generados.

Clase real\Predicción	Galaxias	Estrellas	Cuásares
Galaxias	0.968	0.027	0.005
Estrellas	0.173	0.820	0.007
Cuásares	0.010	0.000	0.990

Cuadro 4.2: *Matriz de confusión de los datos de validación una vez balanceado los datos usando ROS*

4.3. Normalización del espectro

Los valores absolutos para cada banda de los diferentes cuerpos cosmológicos son diferentes (Fig. 4.1). Estas diferencias son significativas, luego cabe la posibilidad de que la red esté comparando los valores de cada banda además de/en lugar de los patrones de los espectros.

De ser ese el caso, causaría problemas de cara al futuro debido a que, de ampliar los valores de entrenamiento y validación en el futuro, deben tener también los mismos valores para cada banda o no podrían ser clasificados. Para comprobar esa hipótesis vamos a comprimir los espectros de cada valor en la base de datos para que esté contenido en el intervalo [0,1].

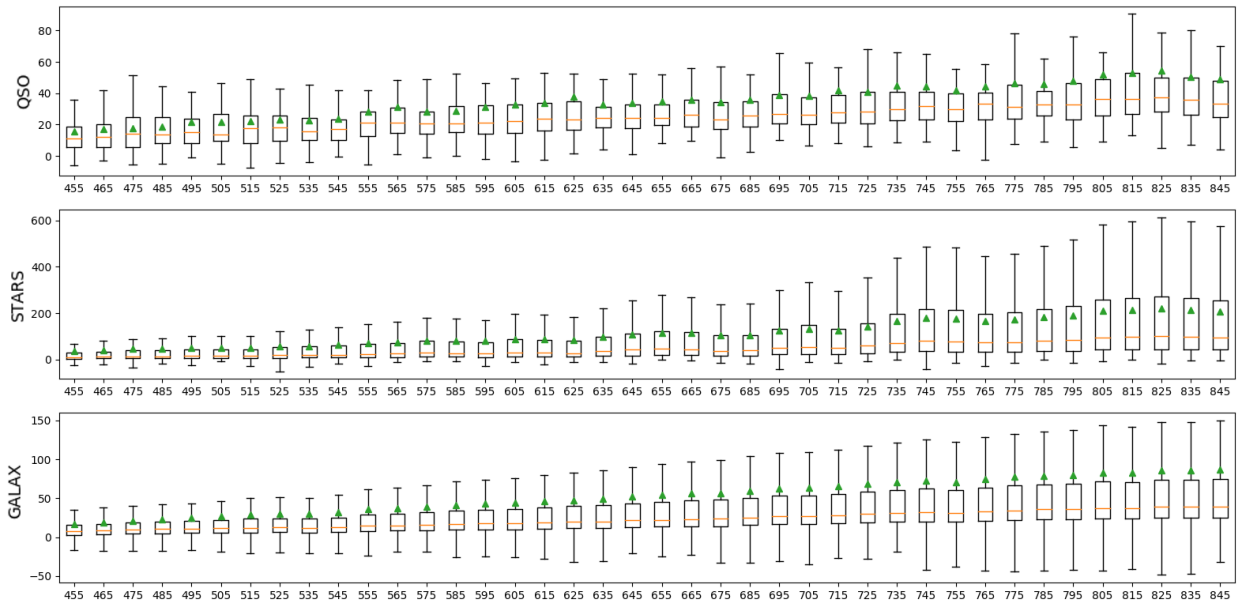


Figura 4.1: *Diagrama BoxPlot de los valores absolutos para los diferentes objetos a clasificar. Nótese que la escala en el eje vertical es diferente para los tres objetos.*

Clase real\Predicción	Galaxias	Estrellas	Cuásares
Galaxias	0.946	0.48	0.006
Estrellas	0.261	0.735	0.005
Cuásares	0.000	0.000	1.000

Cuadro 4.3: *Matriz de confusión de los datos de validación utilizando los espectros normalizados*

La nueva red que obtenemos una vez hemos normalizado y vuelto a aplicar ROS sobre estos datos nos da la siguiente matriz de confusión (cuadro 4.3).

Esta matriz nos confirma que en el caso de que estemos obligando a la red a fijarse en los patrones sigue siendo capaz de generar una red capaz de clasificar galaxias, estrellas y cuásares. Pese a ello, también se puede apreciar cómo se reduce ligeramente la eficacia en la predicción para el caso normalizado. Eso es lo esperado, pues estamos quitando información importante que la red estaba usando y que ayuda a la clasificación.

5. Comparativa de la CNN con otros métodos de clasificación

A la hora de predecir, la red neuronal no es perfecta. Para intentar mejorar dicha precisión, se buscarán otros algoritmos de aprendizaje no supervisado y de clasificación que puedan mejorar la predicción. Los métodos que usaremos son *Density-based spatial cluste-*

ring of applicantions with noise (DBSCAN) y el Análisis de componentes principales (PCA). Como entrada, se usarán las salidas de los filtros de los diferentes espectros, porque esos datos ya han sido tratados previamente por las capas convolucionales.

5.1. DBSCAN

DBSCAN (o agrupamiento espacial basado en densidad de aplicaciones con ruido en español) es un algoritmo no supervisado que busca núcleos de objetos que tengan una alta similitud entre ellos y baja con respecto a los objetos de los demás núcleos. Estas similitudes las calcula operando la distancia entre todos los puntos y después agrupándolos.

Para realizar dicha agrupación, utiliza tres parámetros: el radio de búsqueda para cada punto (ϵ), el mínimo número de puntos necesarios para generar una región densa y la métrica usada (en este caso euclídea). El algoritmo coge un valor aleatorio que no se haya estudiado y busca en un radio todos los puntos vecinos que encuentre. Si supera el número mínimo, genera un núcleo. A partir de este núcleo, siguiendo los mismos pasos busca si aquellos puntos vecinos también formarían un núcleo. Si estos puntos vecinos no lo formasen, se considerarían puntos "densamente alcanzables", aquellos puntos que tienen dentro de un radio ϵ un punto dentro de un núcleo, pero a su vez no alcanza suficientes puntos para formar parte de él. Si un punto no alcanza puntos que formen parte de un núcleo, se consideran puntos anómalos (*outliers*).

Estos objetos anómalos poseen baja similitud con los demás objetos. Puede haber varios objetos anómalos que tengan similitud entre sí, pero no llega a haber suficientes como para formar un núcleo aparte.

Como entrada para este algoritmo usamos las 384 salidas que obtenemos tras ejecutar las capas de convolución sobre el espectro inicial. Estas salidas representan diferentes aspectos que la red neuronal ha considerado importantes de los espectros para separar entre las diferentes clases. Sin embargo, al operar las distancias de más de 30 mil objetos de casi 400 dimensiones pudimos apreciar cómo el tiempo de ejecución se acercaba a los dos meses para un único caso. Es por eso que finalmente redujimos las dimensiones a 32, que son las que se obtienen tras haber pasado por la primera capa oculta de la red neuronal.

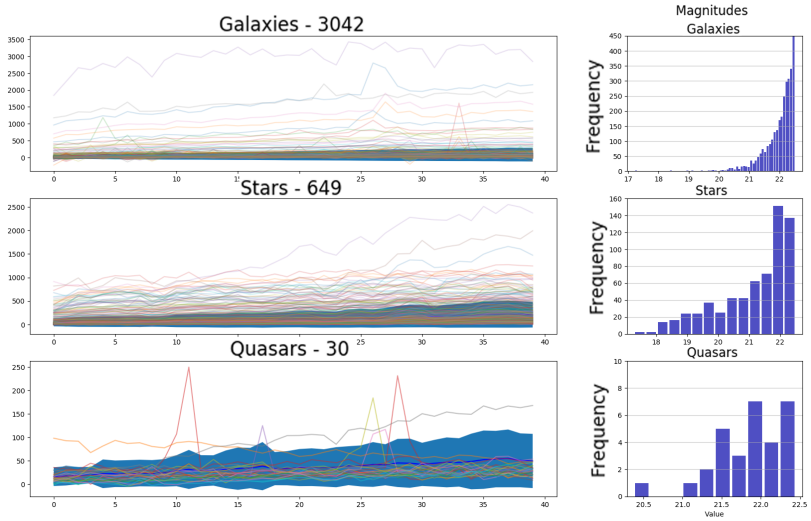
Inicialmente quisimos saber si DBSCAN era capaz de separar entre las diferentes clases. Si fuese así, significaría que se podría seguir alimentando el algoritmo con más datos sin tener la necesidad de obtener sus etiquetas previamente.

El algoritmo nos devuelve un gran núcleo con una gran cantidad de objetos anómalos. Esto es un resultado frecuente para este algoritmo. Es debido a que las separaciones entre las diferentes clases no están limpiamente definidas, y en lugar de clasificar los datos, aísla aquellos valores que tienen una baja similitud con los demás. Estos objetos que difieren de los demás resultan ser aquellos que, o bien tienen un espectro con mucho ruido, o bien tienen un espectro singular que hace que destaquen de entre los demás.

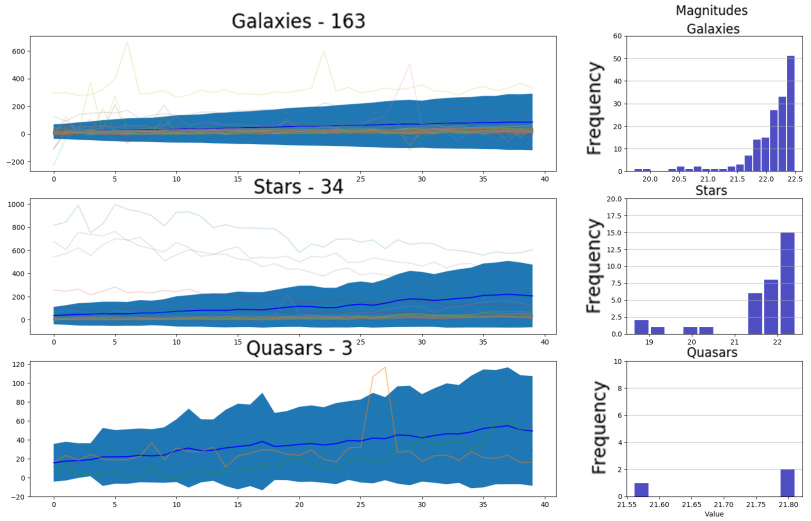
Los parámetros que definen el algoritmo de DBSCAN nos devuelven una mayor o menos cantidad de objetos anómalos en función de lo estricto que se configure el algoritmo. En las siguientes figuras (Fig. 5.1) representamos dos casos específicos: DBSCAN (ϵ=1.75, n=50) identifica un 12 % de los datos como anómalos (Fig. 5.1(a)) y otro menos restrictivo (ϵ=2.5,

n=15) en el que sólo identifica 200 casos (0.6 % del total de observaciones) como anómalos (Fig. 5.1(b)).

En ambos casos podemos observar cómo está separando aquellos objetos que tienen una magnitud alta. Esos objetos son aquellos que tienen una baja relación S/N, además de tener muchos de ellos sus espectros por debajo del valor medio para las tres clases (sombreado azul) (Fig. 5.1). Para el caso con muchos objetos anómalos se puede apreciar que también está separando aquellos puntos que tienen valores muy superiores a la desviación estándar del grupo.



(a) Configuración restrictiva



(b) Configuración poco restrictiva

Figura 5.1: *Espectros y Magnitudes de las dos configuraciones de DBSCAN. Izquierda: configuración estricta, derecha: configuración menos restrictiva.*

Al superponer los histogramas de las magnitudes de los objetos anómalos con el histograma de todos los objetos, identificamos que las distribuciones de los objetos anómalos están desplazadas hacia las magnitudes más altas.

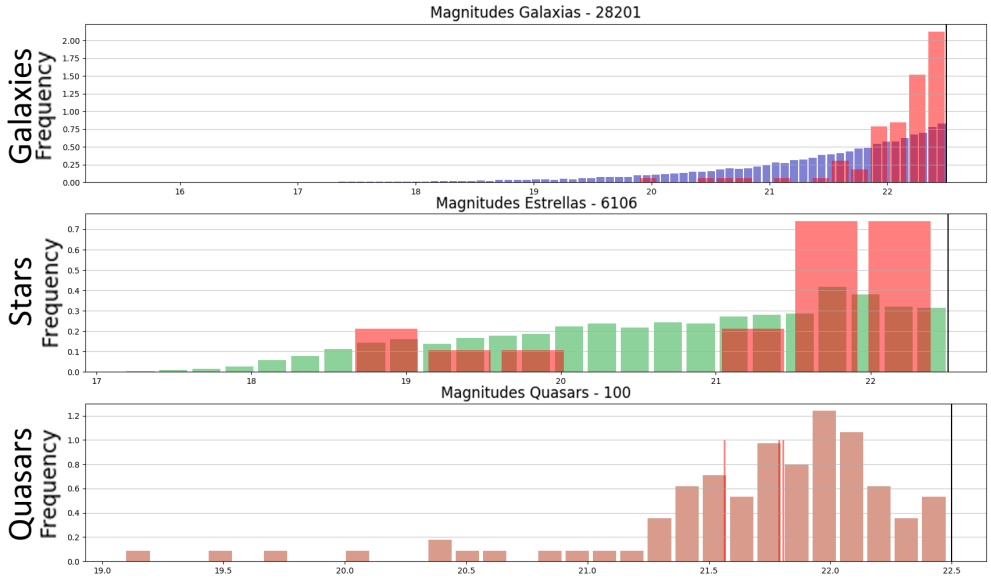


Figura 5.2: *Comparación de los espectros de los objetos anómalos (rojo) con respecto al resto de los espectros.*

Así pues, la utilización de este algoritmo no mejoraría la clasificación de la red a priori.

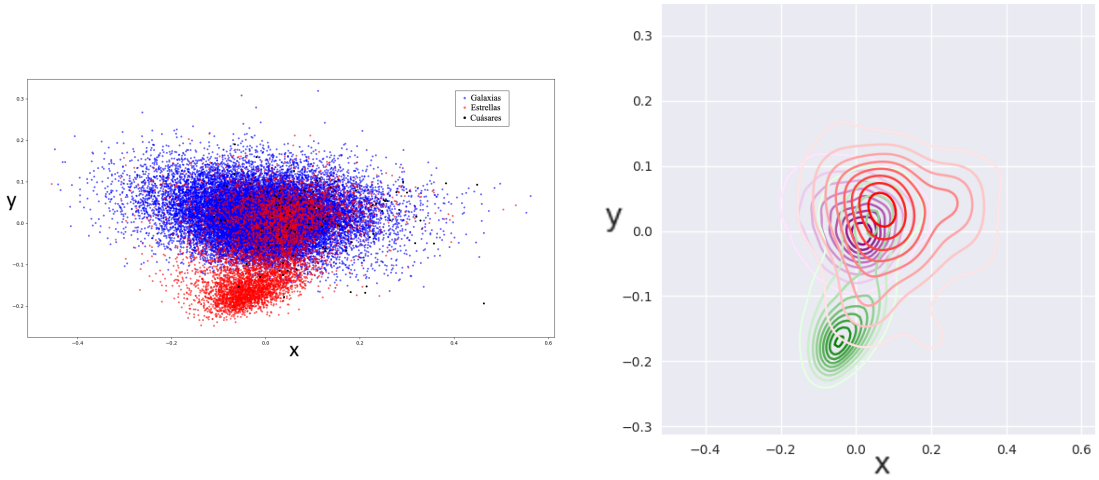
5.2. Análisis de Componentes Principales (PCA)

El Análisis de Componentes Principales es un proceso que se utiliza en muchos ámbitos de la ciencia. El método realiza un cambio de base para minimizar la correlación entre las variables. Posteriormente, se ordenan las nuevas variables según la varianza en los datos originales, permitiendo realizar una reducción de dimensionalidad sobre el conjunto inicial.

Tras realizar el PCA sobre las características (*features*), se pueden truncar las dimensiones a dos manteniendo el 92 % de la variabilidad. Estas dos dimensiones se pueden representar en un diagrama de dispersión y uno de contornos (Figs. 5.3).

La clasificación de objetos mediante PCA consiste en encontrar cortes en los diagramas que se generan en función de los puntos. Al separar por clases, se puede identificar en el diagrama las diferentes poblaciones que se generan en este diagrama. Esto es consistente con el resultado de DBSCAN, pues las diferentes poblaciones no están claramente aisladas. Esto se debe a que las estrellas tienen una distribución bimodal (Fig. 5.4(a)).

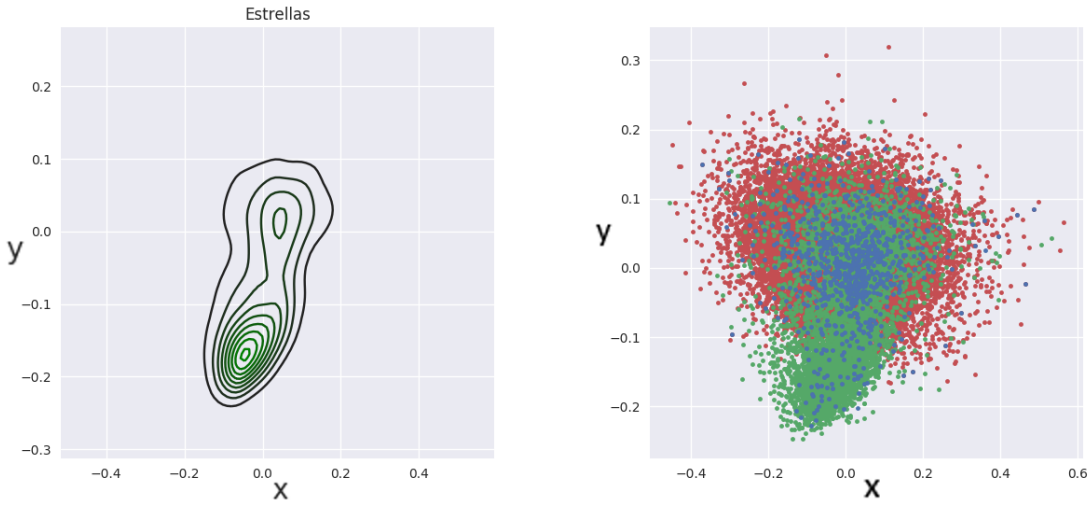
Este resultado nos indica cómo PCA es capaz de diferenciar una gran parte de las estrellas de las galaxias, pero hay un grupo de estrellas que PCA no las diferencia por el espectro de las galaxias. Al comparar la clasificación de estrellas por la red neuronal con los resultados de PCA (Fig. 5.4(b)) se aprecia cómo las estrellas del lóbulo inferior también se pueden clasificar correctamente. Adicionalmente, la CNN es capaz de identificar cerca del



(a) Representación de los puntos estudiados. Azul: galaxias, rojo: estrellas, negro: cuásares

(b) Comparación usando contornos. Morado: galaxias; Verde: esterllas; Rojo: cuásares

Figura 5.3: *Representación de los resultados de PCA, mostrando los dos ejes con mayor variabilidad.*



(a) Diagrama de contornos de la distribución de estrellas al usar PCA.

(b) Comprobación de clasificación de las estrellas que se clasifican mal usando CNN. Rojo: galaxias, Verde: estrellas bien clasificadas, Azul: estrellas mal clasificadas

Figura 5.4: *Identificación de estrellas y su clasificación.*

70 % de aquellas estrellas que no son distinguibles al usar PCA. Pese al parecido que tienen determinadas estrellas en su espectro con las galaxias, la CNN es capaz de diferenciarlas.

6. Utilización de los datos de la morfología junto a la CNN para optimizar la clasificación de objetos cosmológicos

6.1. Clasificación usando morfología de Sloan Digital Sky Survey

Se pueden intentar clasificar los objetos estudiados mediante su valor de extensión usando el catálogo Sloan Digital Sky Survey (SDSS). Siguiendo los pasos mencionados en la sección 3.1, realizamos el cálculo de la concentración restando a la magnitud del modelo el valor de la función PSF. Para el catálogo SDSS, el valor de la frontera es de -0.145. Si la morfología de un objeto supera dicho valor, ese objeto se considerará una estrella, si es menor, se considerará una galaxia.

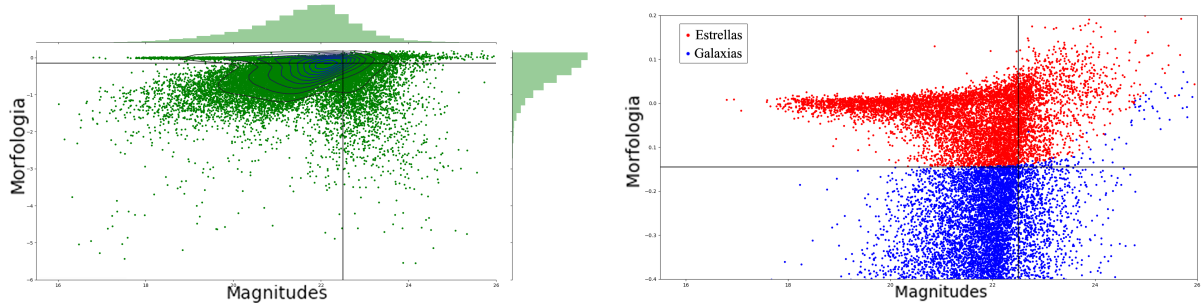


Figura 6.1: *Extensión de los objetos con respecto a su magnitud.*

Al representar la extensión con respecto a la morfología (Fig. 6.1), se puede apreciar cómo la separación entre estrellas y galaxias no es una línea clara. Esto es debido en parte al *seeing* atmosférico debido a usar telescopios terrestres. Para reducir el error en las etiquetas, hemos usado los resultados del telescopio espacial HST por su gran resolución espacial y a no tener el efecto *seeing*. Podemos comparar ambos resultados de clasificación por morfología tomando los datos del HST como correctos cuadro 6.1.

Esta discrepancia en los resultados se puede comprobar usando las etiquetas del HST para diferenciar objetos al representar la extensión respecto a la magnitud (Fig. 6.2). Como se puede observar, para telescopios terrestres, el error que aparece debido al seeing dificulta la clasificación entre galaxias y estrellas usando exclusivamente la concentración de luz del objeto estudiado.

HST\Morfología	Extenso	Puntual
Galaxias	0.8375	0.1625
Estrellas	0.2102	0.7897

Cuadro 6.1: *Comprobación de la eficacia de usar la extensión como clasificador para los datos de SDSS.*

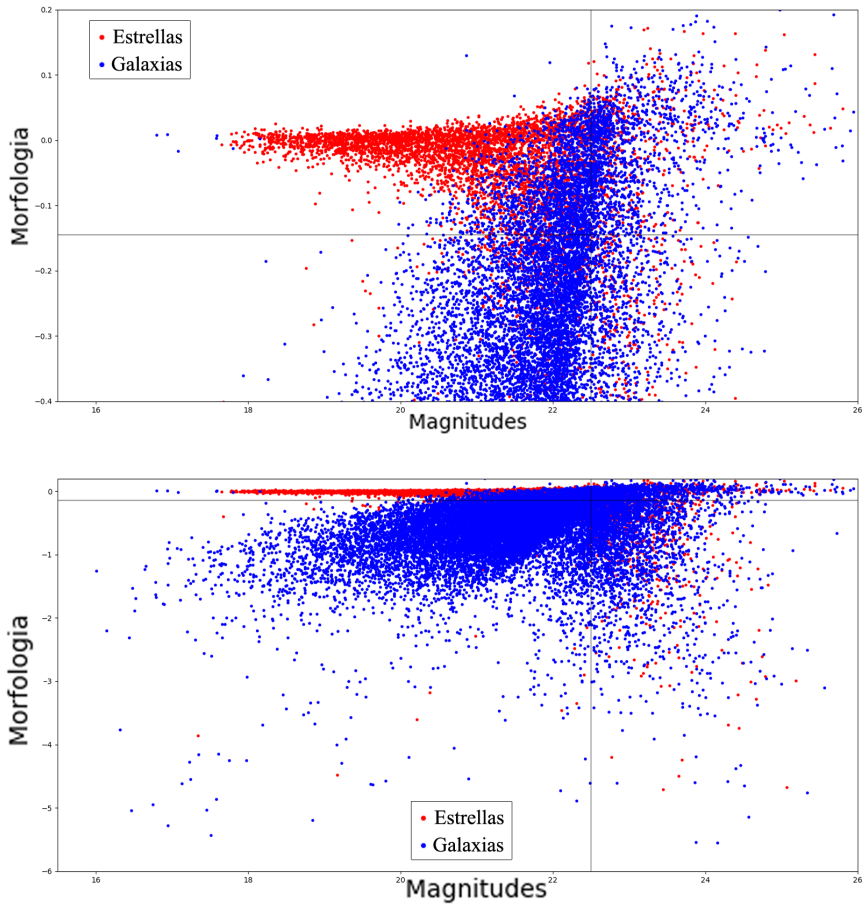


Figura 6.2: *Espectros y Magnitudes de las dos configuraciones de DBSCAN. Izquierda: configuración estricta, derecha: configuración menos restrictiva.*

6.2. Red con morfología

La red CNN ha dado resultados satisfactorios para clasificar entre las tres clases de objetos cosmológicos. No es un método perfecto (cuadro 4.2), pero mejora los resultados obtenidos comparándolos con determinados datos morfológicos como los de SDSS. Los intentos de mejorar la red neuronal usando métodos como DBSCAN o PCA no han representado una mejora sustancial.

La disponibilidad de datos morfológicos de diferentes catálogos nos permite generar una CNN combinando valores de espectros con la morfología obtenida de diferentes catálogos.

Esta combinación se puede realizar con proyectos con una S/N parecida a nuestros valores (SDSS) o de otros proyectos con una S/N mucho mayor (usaremos los datos obtenidos con la cámara Hyper Suprime Cam (HSC) [Aihara et al.[1]]).

Sin embargo, no se pueden añadir los datos morfológicos directamente a la CNN como si fueran una banda extra. De hacerlo así, las capas convolucionales tratarían esos datos como una banda más, intentando encontrar un patrón junto con las bandas adyacentes. Si los datos morfológicos proceden de una fuente muy precisa, añadir dichos datos desde el principio harían que la red ignore los espectros; haciendo que ya no se pueda identificar los cuásares. De ser los datos morfológicos imprecisos, al añadirlos desde el principio aumentaría el ruido del sistema.

Una opción para combinar ambos datos de forma no destructiva sería generar la CNN descrita en la sección 4 y posteriormente juntar las salidas de esa red con una segunda red neuronal que tenga en cuenta la morfología. Esta opción no es recomendable debido a que las dos redes se entrenarían de forma independiente.

La solución que utilizaremos es una sola red con dos entradas escalonadas. En la primera capa, la red recibirá los valores de las bandas. Una vez ha generado la salida categórica, juntará esa salida con el parámetro morfológico y generará un nuevo valor categórico que será la salida final de la red. Esta nueva red entrenará todas sus partes simultáneamente. La estructura de dicha red se puede ver en la figura 6.3. La estructura dentro de las líneas punteadas corresponde a la estructura que teníamos anteriormente de la red convolucional.

Al usar datos con la misma S/N, en Cabayol et al.[4] vieron que la red tiene un resultado claramente mejor que usando la extensión. La combinación de las bandas de PAUCam con la morfología de SDSS resulta en una nueva red que predice peor que simplemente catalogando con la CNN [Datos no mostrados]. En su lugar, utilizaremos los datos del HSC [Aihara et al.[1]]. Estos datos tienen un formato booleano que indica si el objeto es o no extenso. La comprobación de estos resultados con respecto a las etiquetas del HST se pueden encontrar en el cuadro 6.2.

Al generar la nueva red convolucional conjunta los resultados son muy buenos (cuadro 6.3). Esta nueva red neuronal conjunta genera unos resultados superiores a ambos métodos por separado.

HST\Hyper Suprime Cam	Extenso	Puntual
Galaxias	26324	19
Estrellas	4045	224
Cuásares	19	51

Cuadro 6.2: *Eficacia en la predicción para los datos de HSC.*

6.3. Curvas de Pureza - *Positive Predictive Value* (PPV)

Para comprobar la eficacia de la nueva red con morfología, hemos analizado la capacidad predictiva de esta nueva red combinada usando las curvas de pureza (PPV) en función de la magnitud. Estas curvas comparan la eficacia en la predicción a medida que va aumentando la magnitud y el ruido (Figs. 6.4).

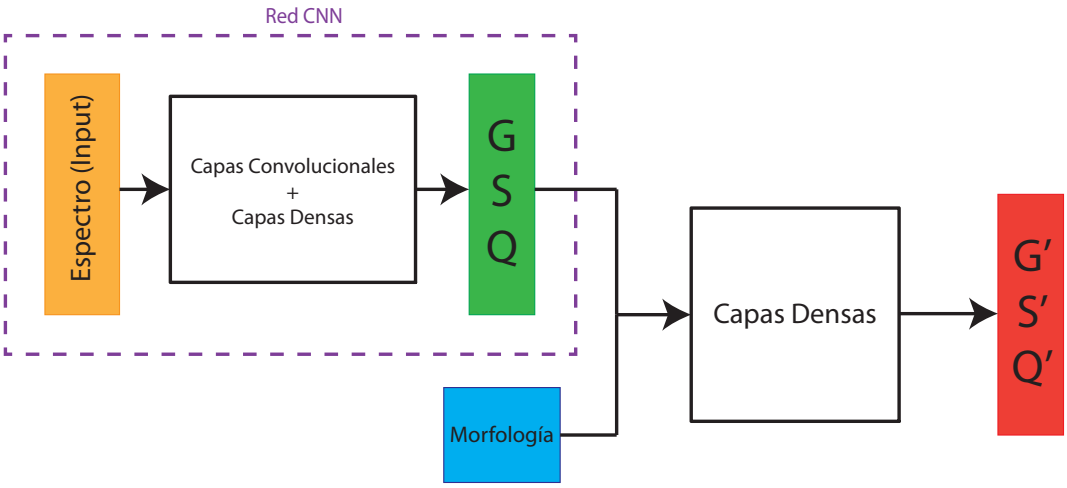


Figura 6.3: *Estructura de la red neuronal conjunta.*

HST\Red Conjunta	Galaxias	Estrellas	Cuásares
Galaxias	0.994	0.001	0.005
Estrellas	0.027	0.965	0.008
Cuásares	0.000	0.000	1.000

Cuadro 6.3: *Matriz de confusión de la red conjunta.*

El uso de la red con morfología tiene dos casos claros. Para las galaxias, la morfología es un método muy bueno para clasificarlas, la red combinada empeora ligeramente esos resultados. Sin embargo, las estrellas de baja magnitud ($<$20mag) se clasifican mejor con la red neuronal sola que con la morfología a secas. A esas magnitudes la combinación de ambos métodos proporciona una precisión mejor que los otros dos métodos.

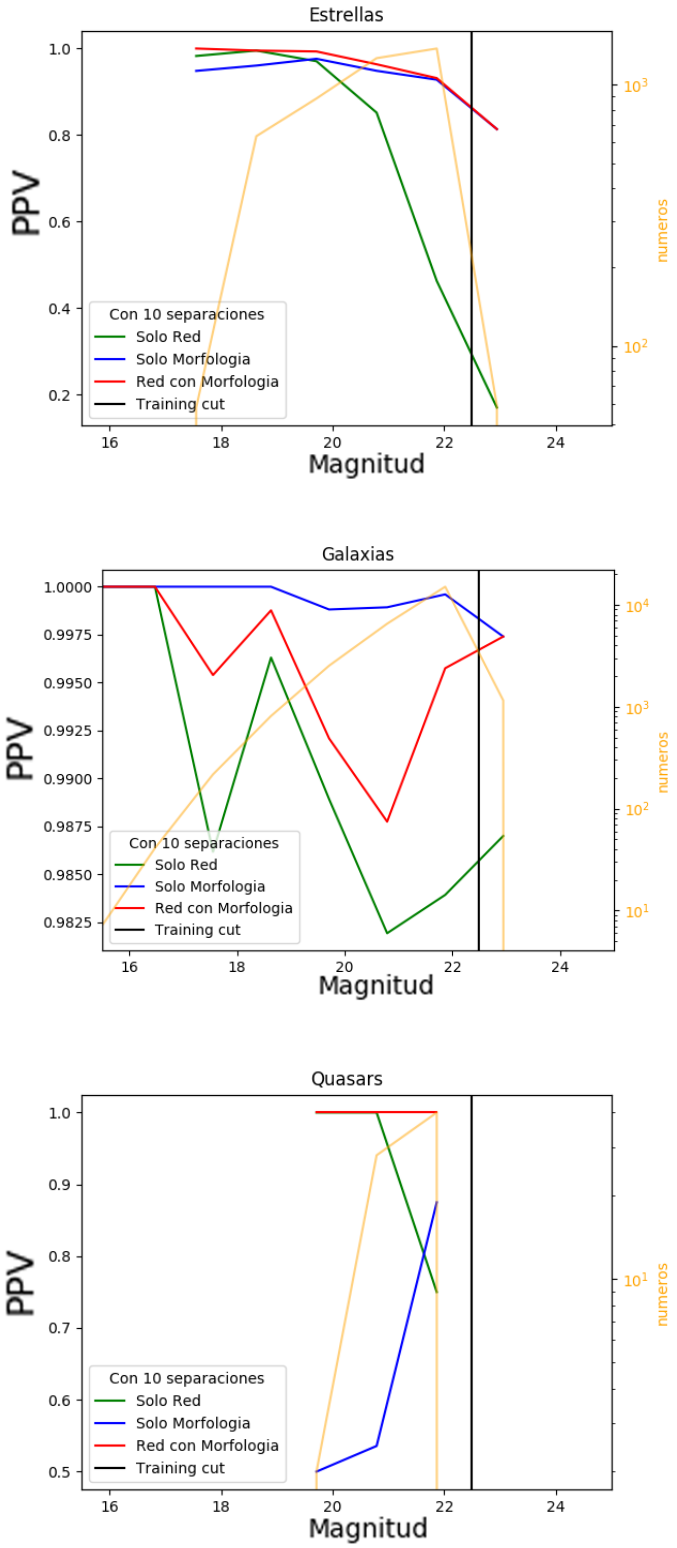


Figura 6.4: *Curvas PPV en función de la magnitud para las estrellas, galaxias y cuásares. Nótese que las escalas son diferentes para los tres casos.*

7. Conclusiones

En este trabajo se ha estudiado diferentes técnicas para mejorar el sistema de clasificación previamente explorado en Cabayol et al.[4] y las dificultades técnicas que eso supone. El cambio más significativo que se ha realizado consiste en ampliar la clasificación para poder aceptar adicionalmente cuásares.

Esta ampliación a su vez conllevaba el problema adicional de tratar con datos muy desbalanceados. Estos datos contienen muy pocos cuásares, que dificultan para la red reconocer dichos patrones. Tras probar con varios algoritmos de corrección, se dicidió quedarse con el algoritmo *Random Over-sampling*, pese a que genere un sobreajuste para estos objetos. De no realizar este ajuste, la red no sería capaz de reconocer los cuásares de entre los demás objetos.

Respecto al sistema utilizado, se ha comprobado que las redes neuronales convolucionales están analizando los espectros de los diferentes objetos mediante la normalización de la red. Realizar eso reduce la cantidad de información disponible, pese a ello sigue siendo capaz de catalogar las diferentes categorías, empeorando ligeramente la predicción.

El otro aspecto que se ha analizado es la comparación de las redes neuronales convolucionales con respecto a otros algoritmos, uno de aprendizaje no supervisado (DBSCAN) y el otro de clasificación (PCA). DBSCAN no funciona como un método de clasificación, pero sí encuentra objetos anómalos. PCA por el otro lado sí es capaz de realizar una ligera clasificación, encontrando un pequeño grupo de estrellas que se pueden diferenciar del resto de objetos analizados.

El útlimo aspecto explorado en este trabajo consiste en la combinación de espectros y morfología para la clasificación de objetos. La realización de una red conjunta de datos de entrada (Fig. 6.3) ha permitido una mejora significativa de la precisión de la red resultante.

Adicionalmente, han quedado dos caminos como posibles trabajos futuros:

1. La primera opción consiste en usar DBSCAN para separar entre objetos del núcleo y objetos anómalos, para posteriormente generar dos redes neuronales para cada uno de los diferentes grupos. Estas redes se centrarían en los aspectos importantes de cada grupo en lugar de intentar ajustarlos todos.

2. La segunda opción consiste en combinar la CNN espectral generada en este trabajo y combinarla con otras CNN que usen la morfología de la imagen del objeto como en Kim and Brunner[10].

Bibliografía

[1] Hiroaki Aihara, Yusra AlSayyad, Makoto Ando, Robert Armstrong, James Bosch, Eiichi Egami, Hisanori Furusawa, Junko Furusawa, Andy Goulding, Yuichi Harikane, Chiaki Hikage, Paul T. P. Ho, Bau-Ching Hsieh, Song Huang, Hiroyuki Ikeda, Masatoshi Imanishi, Kei Ito, Ikuru Iwata, Anton T. Jaelani, Ryota Kakuma, Kojiro Kawana, Satoshi Kikuta, Umi Kobayashi, Michitaro Koike, Yutaka Komiyama, Xiangchong Li, Yongming Liang, Yen-Ting Lin, Wentao Luo, Robert Lupton, Nate B. Lust, Lauren A. MacArthur, Yoshiki Matsuoka, Sogo Mineo, Hironao Miyatake, Satoshi Miyazaki, Surhud More, Ryoma Murata, Shigeru V. Namiki, Atsushi J. Nishizawa, Masamune Oguri, Nobuhiro Okabe, Sakurako Okamoto, Yuki Okura, Yoshiaki Ono, Masato Onodera, Masafusa Onoue, Ken Osato, Masami Ouchi, Takatoshi Shibuya, Michael A. Strauss, Naoshi Sugiyama, Yasushi Suto, Masahiro Takada, Yuhei Takagi, Tadafumi Takata, Satoshi Takita, Masayuki Tanaka, Tsuyoshi Terai, Yoshiki Toba, Hisakazu Uchiyama, Yousuke Utsumi, Shiang-Yu Wang, Wenting Wang, and Yoshihiko Yamada. Second Data Release of the Hyper Suprime-Cam Subaru Strategic Program. *arXiv e-prints*, art. arXiv:1905.12221, May 2019.

[2] I. K. Baldry, A. S. G. Robotham, D. T. Hill, S. P. Driver, J. Liske, P. Norberg, S. P. Bamford, A. M. Hopkins, J. Loveday, J. A. Peacock, E. Cameron, S. M. Croom, N. J. G. Cross, I. F. Doyle, S. Dye, C. S. Frenk, D. H. Jones, E. van Kampen, L. S. Kelvin, R. C. Nichol, H. R. Parkinson, C. C. Popescu, M. Prescott, R. G. Sharp, W. J. Sutherland, D. Thomas, and R. J. Tuffs. Galaxy And Mass Assembly (GAMA): the input catalogue and star-galaxy separation. , 404(1):86–100, May 2010. doi: 10.1111/j.1365-2966.2010.16282.x.

[3] E. Bertin and S. Arnouts. SExtractor: Software for source extraction. , 117:393–404, Jun 1996. doi: 10.1051/aas:1996164.

[4] L. Cabayol, I. Sevilla-Noarbe, E. Fernández, J. Carretero, M. Eriksen, S. Serrano, A. Alarcón, A. Amara, R. Casas, F. J. Castand er, J. de Vicente, M. Folger, J. García-Bellido, E. Gaztanaga, H. Hoekstra, R. Miquel, C. Padilla, E. Sánchez, L. Stothert, P. Tallada, and L. Tortorelli. The PAU survey: star-galaxy classification with multi narrow-band data. , 483(1):529–539, Feb 2019. doi: 10.1093/mnras/sty3129.

[5] Francisco J. Castander, Otger Ballester, Anne Bauer, Laia Cardiel-Sas, Jorge Carretero, Ricard Casas, Javier Castilla, Martin Crocce, Manuel Delfino, Martin Eriksen, Enrique Fernández, Pablo Fosalba, Juan García-Bellido, Enrique Gaztañaga, Ferran Grañena, Carles Hernández, Jorge Jiménez, Luis López, Pol Martí, Ramon Miquel, Christian Neissner, Cristobal Padilla, Cristobal Pío, Rafael Ponce, Eusebio Sanchez, Santiago Serrano, Ignacio Sevilla, Nadia Tonello, and Juan de Vicente. The PAU camera and the PAU survey at the William Herschel Telescope. In , volume 8446 of *Society of*

Photo-Optical Instrumentation Engineers (SPIE) Conference Series, page 84466D, Sep 2012. doi: 10.1117/12.926234.

[6] Milliquas Catalog. Milliquas catalog. URL https://heasarc.gsfc.nasa.gov/W3Browse/all/milliquas.html.

[7] E. W. Flesch. VizieR Online Data Catalog: The Million Quasars (Milliquas) catalog (6.3) (Flesch, 2019). *VizieR Online Data Catalog*, art. VII/283, Jul 2019.

[8] Gaia Collaboration. The Gaia mission. , 595:A1, Nov 2016. doi: 10.1051/0004-6361/201629272.

[9] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016. http://www.deeplearningbook.org.

[10] Edward J. Kim and Robert J. Brunner. Star-galaxy classification using deep convolutional neural networks. , 464(4):4463–4475, Feb 2017. doi: 10.1093/mnras/stw2672.

[11] Felix Last, Georgios Douzas, and Fernando Bação. Oversampling for imbalanced learning based on k-means and SMOTE. *CoRR*, abs/1711.00837, 2017. URL http://arxiv.org/abs/1711.00837.

[12] Alexie Leauthaud, Richard Massey, Jean-Paul Kneib, Jason Rhodes, David E. Johnston, Peter Capak, Catherine Heymans, Richard S. Ellis, Anton M. Koekemoer, Oliver Le Fèvre, Yannick Mellier, Alexandre Réfrégier, Annie C. Robin, Nick Scoville, Lidia Tasca, James E. Taylor, and Ludovic Van Waerbeke. Weak Gravitational Lensing with COSMOS: Galaxy Selection and Shape Measurements. , 172(1):219–238, Sep 2007. doi: 10.1086/516598.

[13] Honglak Lee, Roger Grosse, Rajesh Ranganath, and Andrew Y Ng. Convolutional deep belief networks for scalable unsupervised learning of hierarchical representations. In *Proceedings of the 26th annual international conference on machine learning*, pages 609–616. ACM, 2009.

[14] P. Martí, R. Miquel, F. J. Castander, E. Gaztañaga, M. Eriksen, and C. Sánchez. Precise photometric redshifts with a narrow-band filter set: the PAU survey at the William Herschel Telescope. , 442(1):92–109, Jul 2014. doi: 10.1093/mnras/stu801.

[15] G. R. Ricker. The Transiting Exoplanet Survey Satellite Mission. *Journal of the American Association of Variable Star Observers (JAAVSO)*, 42(1):234, Jun 2014.

[16] G. R. Ricker. The Transiting Exoplanet Survey Satellite (TESS): Discovering Exoplanets in the Solar Neighborhood. In *AGU Fall Meeting Abstracts*, pages P13C–01, Dec 2016.

[17] Ryan Scranton, David Johnston, Scott Dodelson, Joshua A. Frieman, Andy Connolly, Daniel J. Eisenstein, James E. Gunn, Lam Hui, Bhuvnesh Jain, Stephen Kent, Jon Loveday, Vijay Narayanan, Robert C. Nichol, Liam O'Connell, Roman Scoccimarro, Ravi K. Sheth, Albert Stebbins, Michael A. Strauss, Alexander S. Szalay, István Szapudi, Max Tegmark, Michael Vogeley, Idit Zehavi, James Annis, Neta A. Bahcall, Jon Brinkman, István Csabai, Robert Hindsley, Zeljko Ivezic, Rita S. J. Kim, Gillian R. Knapp, Don Q. Lamb, Brian C. Lee, Robert H. Lupton, Timothy McKay, Jeff Munn, John Peoples, Jeff Pier, Gordon T. Richards, Constance Rockosi, David Schlegel, Donald P. Schneider, Christopher Stoughton, Douglas L. Tucker, Brian Yanny, and Donald G. York. Analysis of Systematic Effects and Statistical Uncertainties in Angular Clustering of Galaxies from Early Sloan Digital Sky Survey Data. , 579(1):48–75, Nov 2002. doi: 10.1086/342786.

A. Códigos empleados

Todos los códigos empleados en la elaboración de este trabajo fin de máster se pueden encontrar en un repositorio localizable en:

`https://github.com/enriquegalceran/TFMCNNGSQ`

B. Representación de resultados de la CNN

En Cabayol et al.[4], la red neuronal devolvía una salida binomial que devolvía la confianza que tenía la red de que el valor analizado fuera o no una galaxia. En esta CNN lo hemos cambiado para obtener una salida categórica de tres posibles salidas.

Los tres valores de la salida que tendremos representan la confianza que tiene la red de que sea una galaxia, una estrella o un cuásar respectivamente. La suma de los tres valores debe ser 1 para que represente correctamente una probabilidad, luego se puede representar las respuestas usando diagramas ternarios (Fig. B.1).

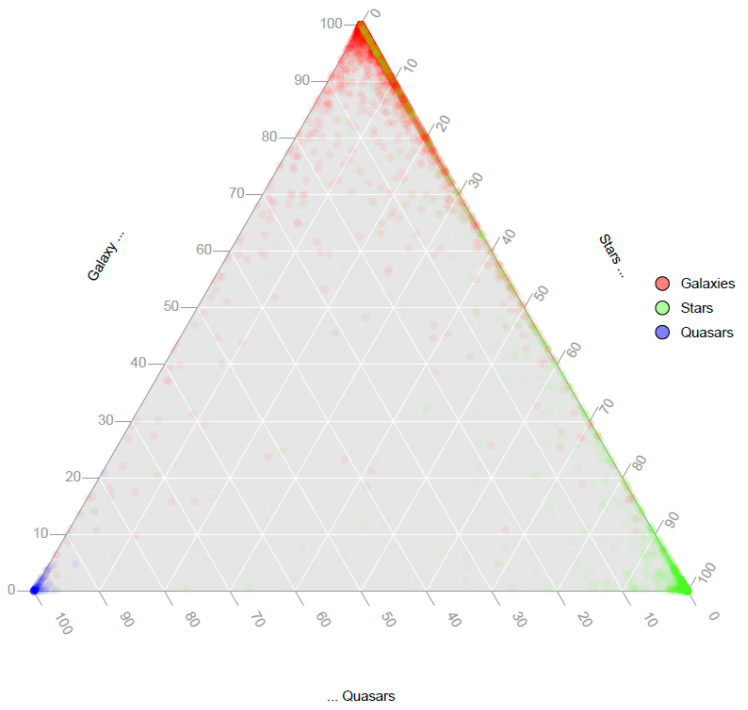


Figura B.1: *Diagrama Ternario representando los resultados de la CNN.*

Se puede apreciar cómo la mayor parte de los objetos se agrupan en las esquinas y las aristas, dejando el centro del triángulo prácticamente vacío. Esto muestra que la red tiene una tendencia a ser capaz de descartar al menos una de las clases.