Universidad de Chile
Facultad de Ciencias Físicas y Matemáticas
Departamento de Ingeniería Eléctrica

Entrega Intermedia

Detección temprana de supernovas en ALeRCE

Integrantes: Diego Castillo
Nibaldo Foix
Profesor: Pablo Estévez
Auxiliares: Ignacio Reyes
Ayudantes: Esteban Reyes
Francisca Cona
Jhon Intriago
Mauricio Romero
Óscar Pimentel
Pablo Montero

Fecha de entrega: 12 de noviembre de 2020
Santiago, Chile

Índice de Contenidos

Lista de Figuras

Lista de Tablas

1. Introducción

ALeRCE: The Automatic Learning for the Rapid Classification of Events es una iniciativa llevaba a acabo por Instituo Milenio de Astrofísica y el centro de modelamiento matemático de la universidad de Chile. Junto a Data observatory y otras universidades e instituciones se ha creado este proyecto con el fin de crear un sistema capaz de detectar, de forma automática, ciertos tipos de eventos astronómicos.

Alerce, en términos técnicos, es un broker de alertas que recibe y procesa observaciones astronómicas provenientes del survey ZTF en tiempo real. Dentro de sus tareas está la clasificación de objetos astronómicos, para lo cual cuenta con distintos algoritmos de *Machine Learning*.

Uno de ellos es el "clasificador temprano", el cual utiliza las imágenes de las primeras detecciones para entregar una clasificación rápida del objeto observado. Es justamente en esta tarea que se entrará en detalle en este proyecto.

La principal motivación de este proyecto radica en la gigantesca cantidad de datos producida cotidianamente en los observatorios astronómicos, la cual, sin técnicas de *Machine Learning* sería imposible de analizar. Al tener un clasificador de objetos astronómicos, como el que se pretende construir, se lograrían identificar fenómenos como las supernovas sin necesidad de que expertos tengan que observar cada imagen obtenida, lo que sería imposible dada la cantidad de datos que se generan.

Para efectos del proyecto los objetivos se dividirán en 4 partes:

1. Familiarización y manipulación de gran cantidad de datos
2. Manipulación inteligente de la información para ejecutar de forma óptima algoritmos de aprendizaje
3. Familiarización e implementación con métodos de redes neuronales convolucionales
4. Evaluación de la red y proposición para mejorar para mejorar el diseño en vista de identificar 5 eventos astronómicos con la más alta tasa de exactitud.

Se espera obtener un modelo capaz de realizar predicción sobre la categoría de las diferentes clases con una exactitud mínima de **resultados esperados** 95 % en general y 85 % para supernovas.

2. Herramientas a Disposición

La base de datos utilizada corresponde a *ALERCE_stamps_2020.pkl*, la cual se compone exclusivamente de un paquete de datos en formato *pkl* proveniente del modulo pickle de python. Este archivo contiene un diccionario con 3 atributos importantes:

- 'Images': Matrices, generalmente cuadradas de 63x63, que contienen 3 muestras para cada valor, representando 3 imágenes para cada objeto astronómico. El primer canal de cada imagen corresponde a "Science", el segundo a "Templatez el tercero a la diferencia entre los dos primeros.
- 'Labels': Número entero que representa la categorización de cada imagen. Se encuentran 5 clases: Active Galactic Nuclei (0), Supernova (1), Variable Star (2), Asteroid (3) y Artifacts (4).
- 'Metadata': Lista de valores que representan información adicional sobre cada imagen.

En términos de software se utilizará el código de programación python ya que es el lenguaje de programación donde viene comprimido la data. Asimismo, python posee varias librerías que servirán para hacer mas ameno el trabajo esperado. Algunas de estas son:

- Pandas: librería para manipular gran cantidad de datos
- numpy: librería para trabajar con tipos de datas array
- matplotlib.pyplot: librería útil para graficar funciones e imágenes
- tensorflow: librería útil para la construcción de redes neuronales

3. Metodología a usar

3.1. Preprocesamiento de Datos

Antes de definir y utilizar cualquier algoritmo de *Machine Learning* es necesario hacer una limpieza de los datos a utilizar. Cualquier tipo de error en los datos conllevará a un error mayor en el proceso de aprendizaje de la máquina.

A fin de concretar lo dicho, se trabajará los siguientes aspectos:

1. Verificar si existe el mismo número de imágenes que de labels y metadata.
2. Verificar si las imágenes son cuadradas. Si no lo son, eliminaras.
3. Verificar cuántos tipos de datos NaN hay y cambiarlos por 0.
4. Acortar las imágenes de 63x63 a 21x21 (centradas) para optimizar el tiempo de ejecución del aprendizaje. Las imágenes siempre están centradas y por lo general los bordes no aportan información.

3.2. Separación en *Sets*:

Para cualquier tipo de algoritmo de aprendizaje, existe la necesidad de separar los datos entregados en grupos de entrenamiento y validación y test. La separación va a tener repercusiones en como aprenderá la red igual que como se podrá medir su rendimiento. En nuestro caso, al menos preliminarmente, se realizarán conjuntos de validación y testing balanceados, extrayendo 300 imágenes de cada clase, de las cuales 200 irán a testing y 100 a validación. Las imágenes restantes quedan en el conjunto de entrenamineto.

3.3. Algoritmo de Detección

Dada la naturaleza del problema se considera que una **red convolucional** podría ser útil a la hora de detectar eventos en imágenes. La naturaleza de convolución pareciera ser efectiva para el problema. Por el hecho de haber sido utilizado en tareas anteriores, se escogerá ese método.

Una red neuronal convolucional es una variación de un perceptrón multicapa, sin embargo, debido a que su aplicación es realizada en matrices bidimensionales. Son muy efectivas para tareas de visión artificial, como en la clasificación y segmentación de imágenes, entre otras aplicaciones.

El principal fuerte de una red convolucional a la hora de detectar imágenes corresponde a su capacidad de aprender patrones, que en el caso de una imagen podrían ser, por ejemplo, la presencia de líneas, bordes, sombras, etc. Una red convolucional logra realizar esto requiriendo solo una pequeña fracción de los parámetros que requeriría una red MLP, ya que, como se ha dicho, logra generalizar patrones en lugar de necesitar aprender cada combinación de pixeles posible de memoria.

Una red convolucional se separa en los siguientes componentes:

1. Entrada
2. Capas convolucionaes
3. Capas de pooling
4. Capas Fully Connected:

Inicialmente se utiliza una red simple, detallada en la tabla 2 de este informe. Posteriormente, se intentará recrear la red utilizada en ."En"hanced Rotational Invariant Convolutional Neural Network for Supernovae Detection"[1] y se probarán modificaciones adicionales a esta. Debido a lo anteriormente explicado, es bastante difícil predecir los recursos computacionales necesarios al final del proyecto.

En lo que concierne el estado actual del proyecto, la red utiliza 149.245 parámetros, lo que equivale aproximadamente a 597kB de almacenamiento, pero sin dudas en estados posteriores del proyecto este almacenamiento aumentará considerablemente.

Además, también en cuanto a la red preliminar que se utiliza para esta entrega, utilizando los ordenadores personales de los integrantes del grupo se aproxima una velocidad de entrenamiento de 1000 evaluaciones cada 30 segundos. De esta manera, en caso de recorrer toda la base de datos, el entrenamiento tardaría 25 minutos aproximadamente. En versiones posteriores de la red se espera una de entrenamiento mucho menor, debido a la mayor complejidad de la red, pero al mismo tiempo se pretende utilizar la GPU brindada por la herramienta Google Colab, lo que aceleraría el procedimiento.

En cuanto a las capas de la red, a lo largo del proyecto se puede cambiar el número de capas para cada tipo de ellas, además de parámetros como la profundidad de cada capa convolucional, el tamaño de cada convolución, el tamaño de cada pooling y la cantidad de unidades de cada capa *fully-connected*.

Por otro lado, los hiperparámetros a tener en consideración son:

- Épocas: la cantidad de épocas a realizar.
- Número de batches: En cuántos conjuntos se dividen los sets de entrenamineto y validación.
- Optimizador: A priori
- Tasa de aprendizaje: Inicialmente se utiliza un valor fijo, pero podría utilizarse un valor adaptativo.
- Data augmentation: se podría utilizar data augmentation, que corresponde a transformaciones lineales (traslaciones y/o proporciones) que se le realizan a los datos ingresados, con el fin de que la red no se aprenda "de memoria"las imágenes.

Otro punto a considerar es la función de costo o error a minimizar en en el algoritmo. Por lo general en redes convolucionales se usa la función de Cross-Entropy o Entropía Cruzada.

$$H(p,q) = -\sum_{x \in X} p(x) log\, q(x)$$

En cuanto a la detención del algoritmo, al ser un clasificador que utiliza batches, se propone que el algoritmo se detenga al acabar todos los datos de entrenamiento o al "saturar.[el] accuracy de validación obtenido. La última condición se cumpliría en caso de que el algoritmo empeore su accuracy de validación una cantidad determinada de veces seguidas, por ejemplo 10.

La evaluación del modelo, en una primera instancia, es realizada obteniendo las medidas de accuracy global y para las supernovas, considerando que se requiera un clasificador con las 5 categorías. Adicionalmente se evalúa un eventual enfoque binario, que en el caso de realizarse, se guiara por una medida F-score, que considere con bastante importancia un buen valor de recall, ya que lo que más se desea evitar es que supernovas queden clasificadas como no-supernovas. Esto último debido a que se ve mucho más factible tener humanos que verifiquen la veracidad solo de las detecciones de supernovas, y descarten los objetos incorrectos a alguien que tenga que encontrar supernovas dentro de las clasificaciones de no-supernovas.

4. Resultados Preliminares

En la presente sección de este trabajo se presentan los avances alcanzados por el grupo a la fecha, los cuales se separan en análisis y preprocesamiento de datos y resultados del algoritmo preliminar.

4.1. Análisis y preprocesamiento de datos

Se encontraron los siguientes resultados para la base de datos:

Tabla 1: Información sobre paquete de datos

Cantidad de Ïmages"	52.244
Cantidad de "Labels"	52.244
Cantidad de "Metadata"	52.244
Cantidad de Imagenes no cuadradas	650
Cantidad de valores "NaN"	9.743.759

La primera observación obtenida de la base de datos es el número de instancias presentes en esta, el cual es de 52.244 para sus tres atributos, por lo que se comprueba que no hay errores en las dimensiones de estos.

Los objetos astronómicos presentes en la base de datos siguen la distribución presentada en la figura 1, donde se observa un claro desbalance, sobre todo concerniente al número de supernovas de la base de datos.

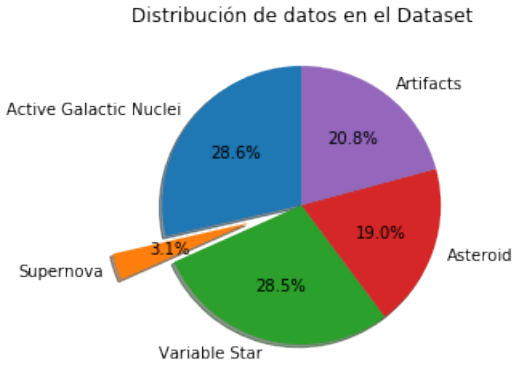

Figura 1: Distribución de Labels

A continuación se observan ejemplos de las imágenes contenidas dentro de la base de datos.

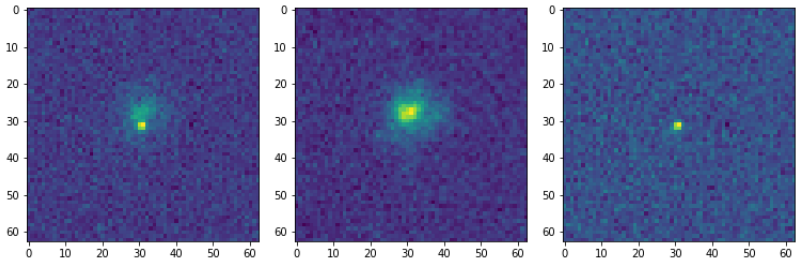

Figura 2: Ejemplo de supernova

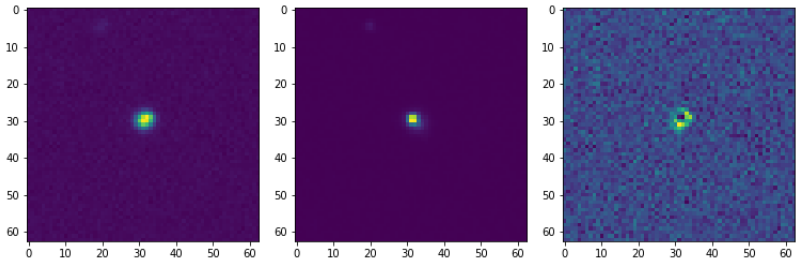

Figura 3: Ejemplo de estrella variable

En cuanto a las imágenes no cuadradas, se han encontrado 650 de estas, las cuales siguen la distribución presentada en la figura 4:

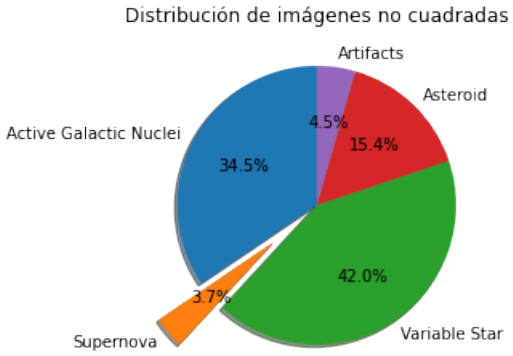

Figura 4: Distribución de imágenes no cuadradas

Al no haber un gran número de supernovas entre estos objetos (24) y por recomendación del equipo docente, estos objetos son eliminados de la base.

Adicionalmente, como se observa en la tabla 1, se ha encontrado una cantidad significativa de valores indeterminados (NaN), los cuales son satisfactoriamente reemplazados por ceros.

Las imágenes logran ser normalizadas exitosamente.

Finalmente, al recortar las imágenes normalizadas, se obtienen imágenes de 21x21 pixeles. En la figura 5 se observan las imágenes recortadas de la supernova ilustrada previamente (figura 2). Estos recortes son guardados en una

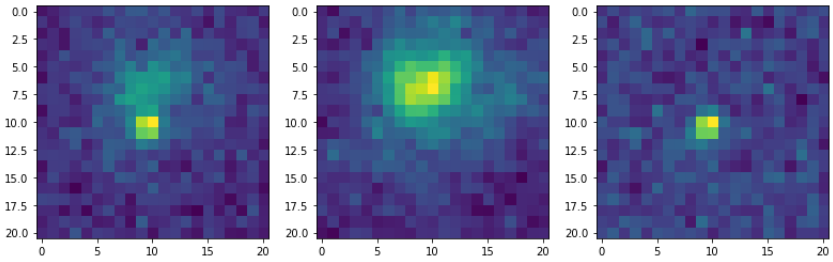

Figura 5: Ejemplo de imágenes de supernova de 21x21 pixeles

Tras la realización del preprocesamiento se obtiene un diccionario con 4 atributos: las imágenes completas normalizadas, las imágenes normalizadas y recortadas, los labels de los objetos y la metadata. Este diccionario se almacena exitosamente en un archivo *pkl*, obteniendo la base de datos preprocesada que se utilizará en adelante.

4.2. Algoritmo

La red convolucional preliminar utilizada se configura de la siguiente manera:

Tabla 2: Estructura del modelo

Layer	**Salida output**	**# params**
Conv16-3	(None, 19, 19, 16)	448
Conv32-3	(None, 17, 17, 32)	4.640
Maxpooling	(None, 8, 8, 32)	0
Conv64-3	(None, 6, 6, 64)	18.496
Conv128-3	(None, 4, 4, 128)	73.856
Maxpooling	(None, 2, 2, 128)	0
Flatten	(None, 512)	0
FC100	(None, 100)	51.300
Dropout	(None, 100)	0
FC5	(None, 5)	505

Para comprender los resultados obtenidos en las matrices de confusión es importante recordar la categorización de los objetos:

0- Active Galactic Nuclei

1- Supernova

2- Variable Star

3- Asteroid

4- Artifacts

Se utilizan los sets de validación y testing descritos en la metodología de este trabajo, los cuales se componen de 500 y 1000 imágenes respectivamente, y ambos son balanceados entre las 5 categorías de objetos.

Inicialmente se utiliza como training set a todos los datos restantes, resultando un conjunto claramente desequilibrado. En este intento se percibe que tras un número determinado de épocas, proporcional al número de batches, la función *loss* se indefine, siendo imposible, por ejemplo, utilizar más de 10 épocas al haber 50 batches.

Considerando el problema descrito anteriormente, la matriz de confusión obtenida con la utilización del conjunto de entrenamiento desbalanceado se observa en la figura 6. Se obtiene un 68,8 % de accuracy total, pero un 0 % de accuracy para supernovas. Para este intento fue utilizada una tasa de aprendizaje de 0,1, 100 batches y 15 épocas.

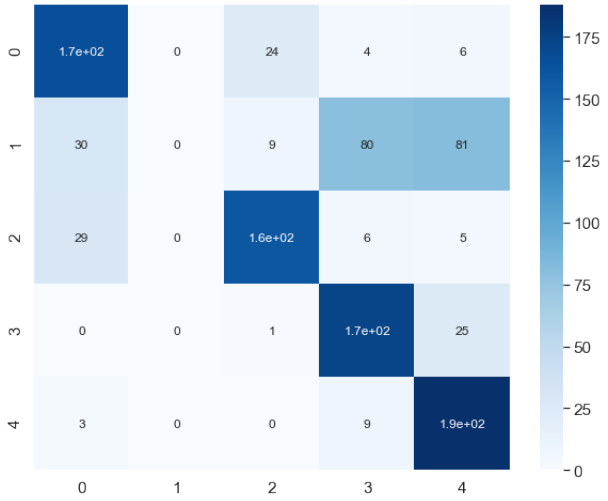

Figura 6: Matriz de confusión inicialmente obtenida

Se realiza un segundo intento, esta vez utilizando un conjunto de entrenamiento balanceado con *undersampling*. La matriz de confusión obtenida se presenta en la figura 7, obteniendo un 60,2 % de accuracy total y un 40,5 % de accuracy para supernovas. Para este intento se dividió el set de entrenamiento en 50 batches y se realizan 10 épocas, además de considerar una tasa de aprendizaje de 0,1.

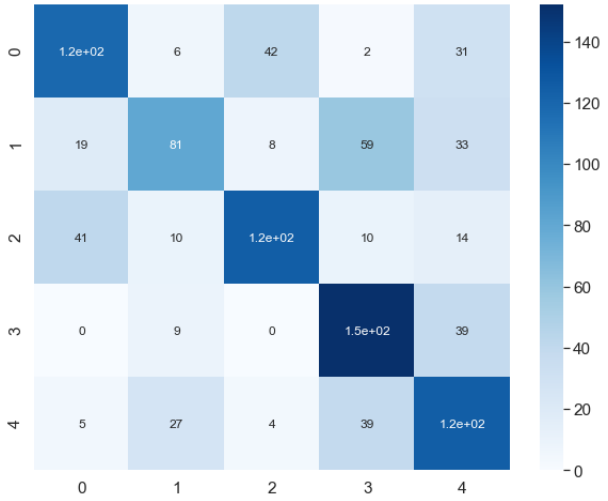

Figura 7: Matriz de confusión de la red utilizada con training set *under-sampled*

Adicionalmente, para este último ensayo se obtienen las siguientes curvas de aprendizaje:

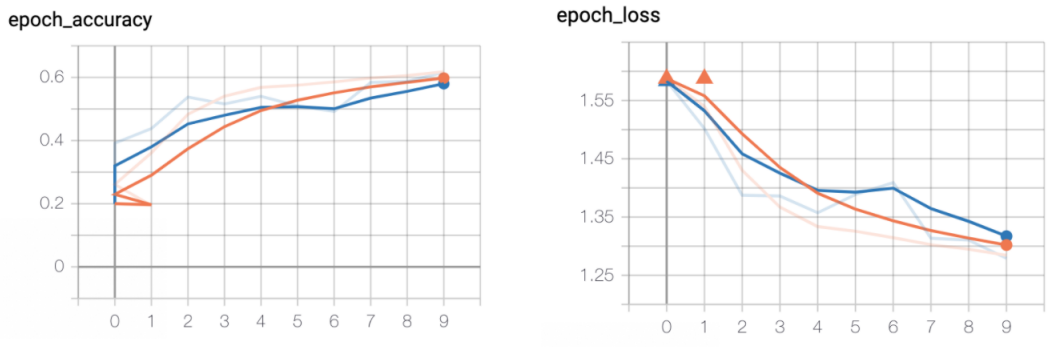

Figura 8: Curvas de aprendizaje (naranjo) y validación (azul) de la red convolucional

5. Análisis

Primero que nada, se observa que existe una gran cantidad de datos que se encuentran en el paquete de datos. En efecto, la tabla 1 muestra que se encuentran 52.244 valores en las 3 categorías. La cantidad de imágenes no cuadradas es poca en relación al total ($\approx 1\,\%$). Tras el recorte de estas, las imágenes totales disponibles para trabajar serán 51.594. La cantidad total de valores NaN era alta y fue necesaria una buena cantidad de procesamiento computacional para procesar. De todas formas esta parte solo se realiza una vez ya que después se guarda como un nuevo archivo pkl.

Posteriormente, se observa una distribución no homogénea de los labels. El grupo "Supernova", el grupo de principal interés posee muy pocas muestras para trabajar. Esto impone una dificultad ya que este es la categoría con la cual se quiere tener buen rendimiento. En el caso más simplificado de categorizador binario, se posee grupo a encontrar: $3.1\,\%$ vs los $96.9\,\%$ restantes. Sin duda esto representará un desafío con los pasos a seguir.

La recomendación de acortar las imágenes parecen funcionar bien. Todas las fotos que se observaron están centradas. como aparecen en Figura 3, 2 y 5. Esto ayudará a aliviar la capacidad computacional necesaria para el mejoramiento de la red. No se descarta, sin embargo, evaluar la red con las imágenes completas y de esta manera comparar los rendimientos para observar cuánto empeora el rendimiento de la red tras el recorte de las imágenes.

Con respecto al algoritmo, se utilizó una configuración inspirada en trabajos anteriores del presente curso, como lo son la actividad de cátedra 5 y la tarea 2. Para esta entrega, más que lograr una configuración óptima, se pretende llegar a un clasificador básico que permita corroborar el prepocesamiento de los datos y familiarizarse con las bibliotecas y comandos para crear redes neuronales.

Se ha evidenciado la necesidad de balancear el conjunto de entrenamiento, idealmente expandiendo los datos de las supernovas (clase menos frecuente) en lugar de cortando los demás datos.

En figura 7 se pueden ver resultados preliminares de la red. Se oberva 84 supernovas detectadas correctamente , mientras que 36 son detectadas como Asteroide y 53 como Artifacts. De esta forma, más de la mitad de las supernovas no son detectadas como tal.

Las otras clases sí tienen una buena exactitud de acierto, por lo que existe un problema solo para la clase que más interesa. Es por eso que se tendrán que proponer alternativas para atacar el problema, como podría ser un clasificador binario, o como se mencionó anteriormente, aumentar el número de imágenes de supernovas artificialmente, utilizando técnicas vistas en clases, las que deben incluir invariabilidad a las rotaciones y traslaciones.

Finalmente, en términos de exactitud y función de pérdida presentes en la figura 8, se observa un comportamiento adecuado. En primera instancia la exactitud sube mientras aumentan las épocas y perdida descienden de la misma manera. En relación a la diferencia entre test y validación tienen formas similares por lo que se descarta overfitting. Cabe destacar que se esta trabajando con pocas épocas por lo que se esperaba una un rendimiento como este.

6. Pasos A seguir

Al haber finalizado el preprocesamiento de datos y teniendo un modelo de red preliminar, se considera que los pasos a seguir más importantes son resolver el déficit de datos de supernovas y perfeccionar el rendimiento de la red realizándole modificaciones a su estructura y a los valores de sus parámetros.

Igualmente, aparecen tareas algo más pequeñas a realizar, como exportar el trabajo a Google Colab.

A continuación se presenta la carta Gantt propuesta por el grupo, la cual incluye las tareas realizadas y a realizar, las distintas semanas del calendario académico y la cantidad de horas de trabajo a estimadas para cada tarea. Además, se incluye el estado de dicha tarea, el cual puede ser realizado.[en] verde, .[en] realización"(naranjo) y a realizar (azul).

	Semana 5	Semana 6	Semana 7	Semana 8	Semana 9	Semana 10	Semana 11	Semana 12	Semana 13
Tareas a realizar	(05/10-09/10)	(12/10-16/10)	(26/10-30/10)	(02/11-06/11)	(09/11-13/11)	(16/11-20/11)	(30/11-04/12)	(07/12-11/12)	(14/12-18/12)
Análisis y preprocesamiento	10 horas	10 horas	5 horas	5 horas					
Separación de sets			5 horas						
Modelo preliminar				5 horas					
Expansión de datos de supernovas					5 horas				
Exportación del trabajo a Google Colab					5 horas				
Reunión resolución de dudas					0,5 horas				
Meeting 3						0,25 horas			
Replicar modelo del paper						10 horas	5 horas		
Meeting 4							0,25 horas		
Proponer y testear mejoras							7 horas	5 horas	
Definición red óptima								3 horas	
Preparación informe final								2 horas	8 horas
Preparación presentación final								5 horas	2 horas

Figura 9: Carta Gantt del proyecto

La expansión de datos de supernovas se refiere a encontrar y realizar algún método de manera que las categorías en el set de entrenamiento sean balanceadas sin hacer *undersampling*.

7. Conclusión

El preprocesamiento de datos se considera finalizado, logrando satisfacer todos los requerimientos pedidos por el equipo docente durante las reuniones tenidas. Este proceso ha permitido la familiarización y manipulación de grandes cantidades de datos, satisfaciendo así uno de los objetivos del proyecto.

Se han definido los pasos a seguir para lograr los objetivos del proyecto, así como los plazos para cada uno en una carta Gantt.

Se ha avanzado satisfactoriamente hasta obtener un modelo de red preliminar, logrando una familiarización con las librerías y funciones necesarias para esto. Sin embargo, este modelo aun obtiene métricas muy lejanas a lo deseado, por lo que se necesita trabajar fuertemente en mejorar el rendimiento del clasificador.

Se cree necesario resolver ciertos problemas observados durante el entrenamiento del modelo preliminar, como la indefinición de la función *loss* tras cierto número de iteraciones y el desbalance entre las categorías en el set de entrenamiento. Para esto se solicitará una reunión con el tutor del grupo.

Se considera necesario trabajar en Google Colab para disminuir el tiempo invertido en los entrenamientos de la red.

Referencias

[1] Reyes, E., Estévez, P. A., Reyes, I., Cabrera-Vives, G., Huijse, P., Carrasco, R., Forster, F. (2018, July). Enhanced Rotational Invariant Convolutional Neural Network for Supernovae Detection. In 2018 International Joint Conference on Neural Networks (IJCNN) (pp. 1-8). IEEE.

[2] Cabrera-Vives, G., Reyes, I., Förster, F., Estévez, P. A., Maureira, J. C. (2017). Deep-hits: Rotation invariant convolutional neural network for transient detection. The Astrophysical Journal, 836(1), 97.

[3] Alerce Science, consultado en Noviembre 2020 http://alerce.science/