

# Reporte 08

Jairo Andres Saavedra Alfonso

Universidad de los Andes

## 1. Objetivos de la semana

- Implementar una red neuronal convolucional para clasificación y regresión.
- Realizar pruebas con capas de Dropout.
- Renormalizar los espectros para entrenamiento.

## 2. Datos

Se utilizaron  $40^3$  datos para todo el proceso de entrenamiento, validación y test. Estos datos están repartidos equitativamente en las cuatro clases principales a clasificar (Estrellas, Galaxias, QSO y QSO BAL). Inicialmente se repartió el conjunto de datos en una relación 80/20 entre *train* y *test*, de los cuales el 25 % de los datos de *train* se asignaron como datos de validación.

Todos los espectros usados tienen una dimensión de 886 píxeles en intensidad. Sin embargo, se reduce a la mitad (443 píxeles) y se usa la otra mitad del espectro como peso para la renormalización. El proceso de renormalización se lleva a cabo mediante la sustracción del promedio y a este resultado se le divide por el RMS de cada espectro, todo esto usando como peso la variabilidad del espectro restante (de 443 a 886 píxeles).

## 3. Estructura de la red neuronal

En esta semana se continuó con la implementación de una red neuronal convolucional con el objetivo de clasificar espectros. La estructura de la red neuronal convolucional se muestra en la siguiente Tabla 1.

**Tabla 1.** Arquitectura tentativa de la red neuronal convolucional.

Capa	Tipo	Característica
Espectro de entrada	Imagen	Espectro con dimensión 1x443 píxeles
conv1	Convolución	Capa de dimensión de entrada 1x443 y salida de 100 con kernel de 10. Stride de 2.
relu	ReLU	Función ReLU
pool	Pooling	Capa de pooling kernel de 2. Stride de 2
conv2	Convolución	Capa de dimensión de entrada 100 y salida de 100 con kernel de 10. Stride de 2.
relu	ReLU	Función ReLU
pool	Pooling	Capa de pooling kernel de 2. Stride de 2
conv3	Convolución	Capa de dimensión de entrada 100 y salida de 100 con kernel de 10. Stride de 2.
relu	ReLU	Función ReLU
pool	Pooling	Capa de pooling kernel de 2. Stride de 2
conv4	Convolución	Capa de dimensión de entrada 100 y salida de 100 con kernel de 10. Stride de 2.
relu	ReLU	Función ReLU
pool	Pooling	Capa de pooling kernel de 2. Stride de 2
dropout	Dropout	Capa de Dropout de 10 %
fc1	Lineal	Capa lineal de dimensión de entrada 1800 y dimensión de salida de 16
relu	ReLU	Función ReLU
dropout	Dropout	Capa de Dropout de 10 %
fc2	Lineal	Capa lineal de dimensión de entrada 16 y dimensión de salida de 4
log_softmax	Función	Función Softmax
Capa de salida	Clase	Capa final de valores 0, 1, 2 y 3 respectivos a las clases (Estrella, Galaxia, QSO y QSO BAL)

Se decidió usar como criterio para calcular la inconsistencia entre los valores predichos y los reales la función de perdida (*Loss Function*) *CrossEntropyLoss*.

#### 4. Rendimiento de la red neuronal

Aun se poseen problemas con la implementación de la arquitectura anteriormente mencionada, por lo tanto no hay resultados sobre el rendimiento de la RNC.

#### 5. Tareas pendientes

- Profundizar sobre el funcionamiento, estructura y evaluación de redes neuronales convolucionales.
- Implementar exitosamente las capas convolucionales.

#### 6. Repositorio Github

En este repositorio se pueden ver todos los reportes semanales y le notebook.  
<https://github.com/MrX1997/Reportes-Proyecto-de-Monograf-a>

## Referencias

1. Busca G., Balland C., (2018).: QuasarNET: Human-level spectral classification and redshifting with Deep Neuronal Networks. Retrieve from: <https://arxiv.org/abs/1808.09955>
2. Paris, I., et al., (2017), Astron. Astrophys...: The Sloan Digital Sky Survey Quasar Catalog: Twelfth data release. , 597, A79