

Reporte 04

Jairo Andres Saavedra Alfonso

Universidad de los Andes

1. Objetivos

- Implementar una red neuronal.
- Observar el proceso de aprendizaje en una red neuronal.

2. Estructura de la red neuronal

En esta semana se implemento, con la ayuda de un tutorial, un red neuronal básica con el objetivo de evidenciar el proceso de aprendizaje en redes neuronales. La estructura de la red neuronal implementada es simple; consiste en una capa de entrada (*input layer*), una capa escondida (*hidden layer*) y una capa de salida (*output layer*).

Inicialmente se escogió aleatoriamente un *set* de 1000 espectros reconocidos como objetos para entrenamiento. Estos flujos poseen una dimensión de 886 píxeles. Por lo tanto, La capa de entrada tiene una dimensión de 886 neuronas las cuales reciben el logaritmo del flujo de los espectros reconocidos. La capa escondida o *hidden layer* posee una dimensión de 16 neuronas, correspondiente a multiplos de 4. Finalmente, la capa de salida tiene dimensión 1000 neuronas.

3. Entrenamiento de la red neuronal

Antes de proceder a entrenar la red neuronal fue necesario realizar un proceso de escalamiento de los datos. La implementación se realizo mediante *Pytorch* con ayuda del modulo de redes neuronales *nn*.

Se creo la clase *Neural_Network* para trabajar el proceso de aprendizaje. En la clase anteriormente mencionada se ajustaron los parámetros correspondientes a las dimensiones del capas, estas dimensiones me determinan las matrices de pesos. También tenemos una función *forward* encargada realizar la convolución de las matrices pesos con nuestra matriz de espectros mediante la función de activación *sigmoid* con el objetivo de obtener nuevos pesos.

Finalmente, se tiene una función *backward* encargada de actualizar y optimizar los pesos para mininizar la perdida con respecto a nuestros pesos en cada entrenamiento. Esta ultima función es vital en el proceso de aprendizaje pues es la que nos indica si estamos o no teniendo evolución con respecto a cada ronda de entrenamiento.

El proceso de entrenamiento se realizo sobre 1000 rondas. Para cada ronda se evaluó el error de lo predicho con respecto las clases existentes de salida mediante la función de perdida.

4. Rendimiento del proceso de aprendizaje.

Con el animo de determinar, independientemente de si lo predicho es correcto o erróneo, si la red neuronal esta aprendiendo o no con los datos de entrada, se evalúa el error mediante la perdida, dada por la función de perdida, en función de el numero de veces que se entreno. Se debe observar que, conforme el numero de entrenamientos aumenta, el error decrece (Ver Figura 1).

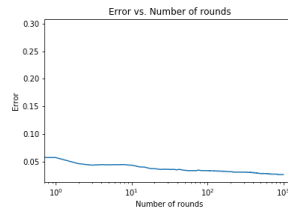


Figura 1. Error vs. Numero de entrenamientos

Se puede concluir que la red neuronal implementada esta aprendiendo con base a los espectros y etiquetas dados.

5. Tareas pendientes

- Profundizar sobre el funcionamiento, estructura y evaluación de redes neuronales.
- Implementar un clasificador exitosamente.

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