

Cloud Atlas

An LstmEncoder for UHECR AirShowers

Gianluca Becuzzi, Lucia Papalini

July 2022

In this project, we propose an example of using deep learning techniques to predict features of UHECR (Ultra High Energy Cosmic Rays) Showers. In particular, when this kind of cosmic ray enters the atmosphere, they produce a particle cascade that can be identified with a set of water-Cherenkov or scintillator ground-based detectors. We aim to train a neural network to predict the height at which the shower formed by feeding it with data on the triggers of such detectors.

The dataset on which we perform the training consists of a simulation of an Auger-like apparatus made of a 9x9 array of detectors. The dataset is given as 80 flattened 9x9 matrices, that correspond to the time series part of the record, and another (flattened) matrix in which each entry is the time of first detected activity, relative to the most intense detector. The overall shape of the array is thus $(80 + 1, 81)$.

The first step is splitting the dataset into the different features: the whole project is based on a custom NumPy *dtype* for storing events' attributes in a single object, sorted by *toa*, *time_series* and *outcome*.

First sight of the dataset showed the need for data augmentation at high values of height, so we built the *Augmentation* class to perform that: it consists of increasing the number of those data by applying rotations and symmetry-flips on the various matrices.

Another important class is *DataFeeder*, and the principle method gives back a *tf.keras.utils.Sequence* with whom we fed the network. It is a way to generate batches of data to avoid memory overload. Being a Keras subclass, it is automatically multiprocessed.

The network we built is a Concatenation of an LSTM (Long Short Term Memory) network for the *time series* and an Encoder network for the *time of arrival*, ended with a series of dense layers that gives back a continuous output for the height of the shower. We built a class for each net module and another class to call them together: *LushlooNet*.

To ensure better performance from the network, we also wrote a module to perform the *hyperparameters tuning* of the network.