

Cloud Atlas

An LstmEncoder for UHECR AirShowers

G. Becuzzi L. Papalini

July 2022

Table of Contents

1 Introduction

2 Preprocessing

3 Neural Network building

Table of Contents

1 Introduction

2 Preprocessing

3 Neural Network building

Questo lo fa la Lush

Dataset, first glance

The dataset is composed of 10^5 simulated events:

- 9x9 grid of detectors
- most intense detector at the center
- 80 frames of time series (40 MHz sampling rate)
- 1 frame of times of first arrival

The single record shape is then $(80 + 1, 81)$

The `pd4ml` package splits by default in 70% train 30% test.

Table of Contents

1 Introduction

2 Preprocessing

3 Neural Network building

Split the dataset

Using a generator (`keras.utils.Sequence`)

- inherit multiprocessing features
- has default callbacks

The dataset is splitted *record by record* for index shuffling

The effect of the high reading time from memory ($\approx 3ms$) is mitigated by `keras` multiprocessing

For the design of the net it is convenient using `numpy` structured arrays

Split the dataset: `funky_dtype`

Data is extracted: from a conceptually *ihomogeneous* list (activity time series together with times of arrival) to
 $(80 + 1, 81) \rightarrow [(\text{"toa"}, (9, 9, 1)), (\text{"timeseries"}, (80, 9, 9))]$
Data can be accessed depending on what is needed

DataFeeder class

DataFeeder class

Augmentation class e amici

Resolution

The reference article suggest using the resolution:

resolution

defined as the standard deviation of the distribution bla bla bla

We point out that

$$\sigma^2 = \frac{1}{N} \sum_i (\delta_i - \bar{\delta})^2$$

is a sensible estimator only if $\bar{\delta} = 0$, for which the adopted resolution is equal to the *RMSE* of the distribution

$$RMSE^2 = \frac{1}{N} \sum_i (x_i - \hat{x}_i)^2$$

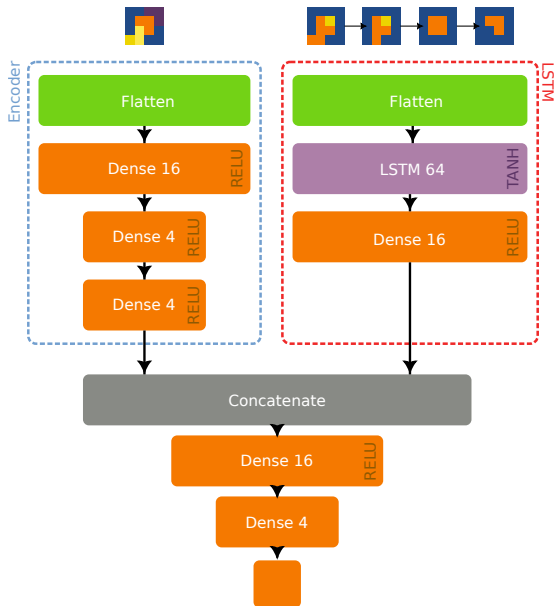
Since (on a typical train) $\bar{\delta} \approx 10\text{m}$ we preferred the RMSE.

Table of Contents

1 Introduction

2 Preprocessing

3 Neural Network building



Overview on the network

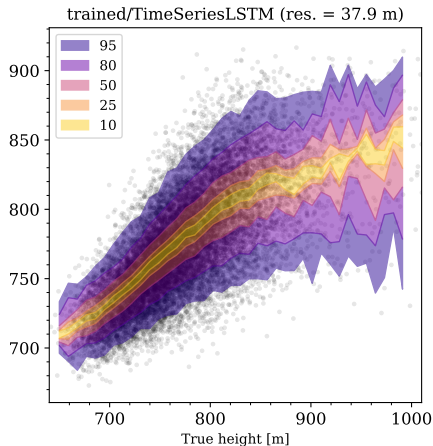
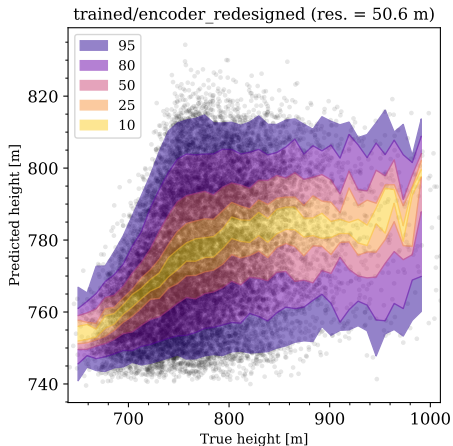
Encoder for time of arrivals

si spiega che cos'è

LSTM for the time series

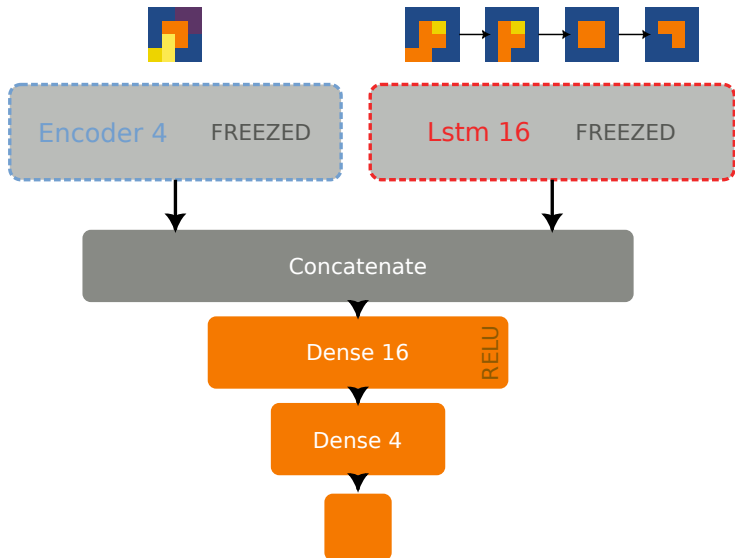
si fa vedere come abbiamo fatto noi

Subnets performance



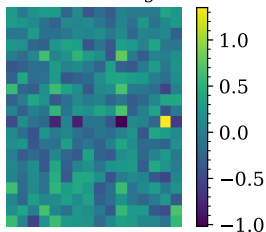
Concatente + dense layers

Subnets train freezing

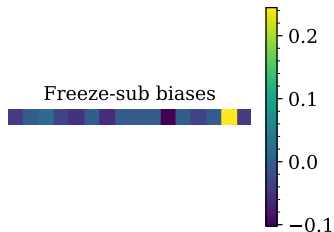


Subnets train freezing

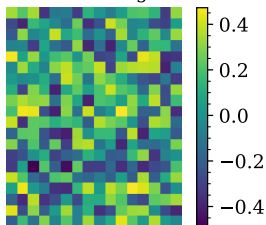
Freeze-sub weights



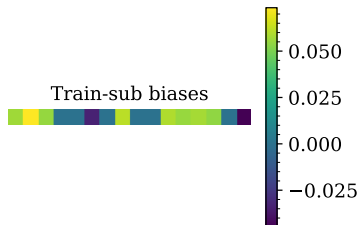
Freeze-sub biases



Train-sub weights



Train-sub biases



Network's output

Hyperparameters tuning

Whole Network performance

Test setup on CircleCI

Danke Schon