Machine Learning replaces Radiative Transfer

Matteo Calafà, Giulia Mescolini, Paolo Motta
Tutor: Dr. Michele Bianco
ML4Science Project in collaboration with the Laboratory of Astrophysics (LASTRO) at EPFL Lausanne, Switzerland

***Abstract*—This paper addresses the task to replace the complex and expensive Radiative Transfer simulations with a Machine Learning approach, in order to study the propagation of radiation in the cosmic Epoch of Reionisation context. The proposed solution includes a Fully Connected Neural Network and a Convolutional Neural Network, the latter being the main focus of the project. In both of them we have applied a physical-based preprocessing of data and the performances are then compared in terms of accuracy.**

I. Introduction and Astrophysical Background

In the cosmic Epoch of Reionisation (EoR), the period in the Universe history of formation of the first galaxies and stars, it is crucial to study how radiation is propagated by astrophysical sources such as galaxies, black holes and stars.
An established method to carry this study out is through Radiative Transfer (RT) simulations, which assign a specific spectral energy density (SED) to each source, compute the propagation of radiation and calculate the absorption/emission coefficient of their surrounding cloud gas.
Although RT is widely employed in many astrophysical domains (such as the study of the radiative feedback of the primordial galaxies/stars in the early Universe, the Supernova explosion and metals contamination, the radiative effect on the morphology of galaxies...) it requires a huge computational effort.
In fact, we are interested in:

- studying a sufficiently **large** cosmological **structure**
- obtaining a simulation with **high resolution**

The number of operations required to solve the equations of these simulations grows exponentially with the number of particles and sources simulated: for example, in the type of simulations considered for this project, the number of operations required scales as $\sim N_s N_p^{\frac{5}{3}}$, where N_s, N_p are the number of sources and particles, respectively.
Therefore, in RT, people are forced to choose between small high-resolution simulations (and resolve the physics down to the interstellar scale) and large but coarse ones (and account for the sample variance of the large cosmic structures).
This is where a Machine Learning approach can enter the game: setting up a Neural Network trained with data coming from a large volume/low resolution RT simulation, we aim at predicting the radiation behavior on a structure with the same width, but higher resolution.

A. Physical Quantities

The propagation of radiation is described by ionization: in the EoR framework, indeed, it is commonly agreed that the first sources start to independently ionise their surrounding neutral gas, creating their so-called ionised bubble or HII regions in a pre-overlap phase ([1]). Continuing to expand the sphere of influence of ionising radiation eventually overlap with nearby companions, such that over time these initially isolated bubbles form a vast interconnected ionised region that stretches until ultimately the entire Universe is fully ionised.
The evolution of the *ionization fraction of hydrogen* x_i is regulated by the following differential equation, and corresponds to its left-hand side:

$$\frac{dn_{HI}}{dt} = -\frac{N_\gamma}{n_{HI}} + \alpha_B(T) n_{HII} n_e$$

This target quantity is a function of:

- n_{src}, the number of the sources per comoving Mpc[1] volume, which corresponds to the first term of the right-hand side of the equation.
- n_{igm}, the density of the intergalactic medium (IGM), given in CGS units. The IGM is the hot, X-ray emitting gas that permeates the space between galaxies, and in the equation it is represented by the second term of the right-hand side.

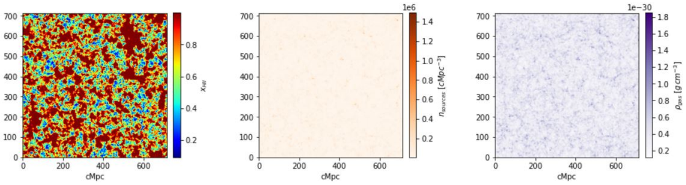


Fig. 1. Values of x_i (left), n_{src} (center) and n_{igm} (right) on a slice of the $3D$ cube that is the project dataset.

II. Data Preprocessing

Our dataset is the same employed in [2]; it consists of $300 \times 300 \times 300$ values in the form $\{(n_{igm}, n_{src}), x_i\}$ for 4 different *redshifts*, i.e. increases in the wavelength of electromagnetic radiation.
For a more intuitive understanding of this concept, we can simply link each redshift with a different moment in the evolution of the Universe.

[1] In astrophysics, it is common to define distances in parsec: $1\,pc = 3.09 \times 10^{13}\,km$

A. Choice of the redshift

First of all, we had to select one redshift to focus on.
By a preliminary analysis of the mean values of the ionization fraction, we obtained the following results:

z	10.11	8.397	7.305	6.905
mean x_i	0.246	0.484	0.753	0.898

In order to avoid problems linked to *class imbalance*, our choice has been for the redshift $z = 8.397$, corresponding to ~ 0.6 Billion Years from the Big Bang.
In this scenario, the division between ionized and not ionized space is nearly $50\% - 50\%$, so both classes are fairly represented.

B. Change of Units of Measurement

The units of measurement used to store n_{igm} and n_{src} are not matching: the former is a density in the CGS unit system, the latter is instead a quantity per comoving mega-parsec. Therefore, we need to perform a transformation:
$[n_{igm}] = \frac{g}{cm^3} = 10^{15}\frac{g}{km^3} = 10^{15} \cdot 3.09^3 \cdot 10^{39}\frac{g}{pc^3} = 2.95 \cdot 10^{55}\frac{g}{pc^3}$
From the above equation we got the right quantity to perform a redefinition of n_{igm}:

$$n_{igm} \rightarrow 2.95 \times 10^{55} \times n_{igm}$$

C. Normalization

The necessity to normalize the quantities n_{igm} and n_{src} comes from the fact that they have extremely different scale (even after the change of units) and variance. Note that x_i is in the range $[0, 1]$ as it represents a fraction, and is therefore not problematic.
For $y = n_{igm}, n_{src}$, indicating with μ and σ the mean value and the standard deviation over the whole $300 \times 300 \times 300$ cube, we have:

$$y_{norm} = \frac{y - \mu}{\sigma}$$

Note that, only for n_{igm}, we have performed a typical transformation in astrophysics, passing to the *cosmic overdensity*:

$$n_{igm} \rightarrow \frac{n_{igm}}{\mu_{igm}} - 1$$

where μ_{igm} is the mean of n_{igm} on the cube.

D. Neighborhoods Generation

The input that the Convolutional Neural Network receives to compute x_i in a given point does not consist of the values of n_{igm} and n_{src} on the whole $300 \times 300 \times 300$ cube, but on a smaller cube.
This choice, which turned out to be determinant for computational effort, is based on an astrophysical principle: there exists a length value, known as *Mean Free Path* (MFP), representing the radius of the volume of influence of each source and intergalactic medium.
Therefore, if we want to obtain the x_i in a given point, we can just focus on the features n_{src} and n_{igm} in its neighborhood.
In our case, the maximum MFP is 57.14 comoving Mpc, and since the resolution of the RT simulation is 2.381 comoving Mpc, the hard limit for the radius of influence corresponds to 24 points.
In the first of our attempts, therefore, the neighborhoods were $49 \times 49 \times 49$ cubes, as we had to consider $24 + 1$ (the central one) $+24$ points in each direction.[2] This size has been changed in the following attempts, and it will be discussed in V-A2.

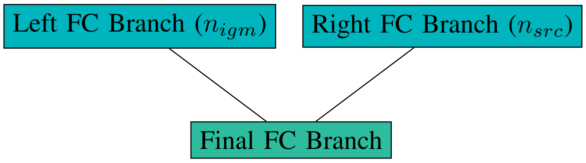
E. Point Sampling

Having chosen to focus on a specific redshift, we have a total of $300 \times 300 \times 300$ points to choose from. To get a sustainable computational usage of resources, we chose to work only with a maximum of 70.000 of them, and we partitioned them into *training set* and *validation set*.
The points were chosen randomly, so that they are widespread over all the $3D$ cube, and we checked that they are sufficiently far from the boundary, so that we could extract a neighborhood in the CNN case.

III. Fully Connected Neural Network

A. Structure

Our first attempt is based on a Fully Connected Neural Network with two dense branches merging into a final dense branch, as it can be seen from the sketch below.



Each branch has three dense layers, which means that each node is connected to *all* the nodes of the previous layer via a weighted edge; this results in a big number of parameters to be learned.
We used the *Exponential Linear Unit* (ELU) as activation function; it is a variant of the RELU, which avoids and rectifies the vanishing gradient problem since it is defined as follows:

$$ELU(z) = \begin{cases} z & z \geq 0 \\ \alpha(e^z - 1) & z < 0 \end{cases}$$

In order to reduce overfitting, and to do a sort of "ensemble averaging" as well, we included a **Dropout Layer** with dropout probability $p = 0.2$; this means that at each training step, we retained with probability $1 - p$ each node.
At first, we included a final sigmoid activation function as well, but it turned out to be ineffective.

IV. Convolutional Neural Network

A. Motivation

As it will be remarked in V-A1, the strategy based on the FNN showed a good performance for medium-high values of x_i, but it lacked in predicting the right x_i for lower values.
This is due to the fact that some points may have a significant

[2] For the sake of lightening the computational effort, the neighborhoods are extracted and stored in a separate script.

x_i even if the n_{src} and the n_{igm} in that point are small. What determines the ionization, in this case, is the closeness to points with high n_{src} and/or n_{igm}.

This lack for the FNN case is due to the fact that it does not take into account the position of the points, therefore the influence of the neighbors is lost. Therefore, we now propose a solution based on a Convolutional Neural Network.

B. Structure

The architecture proposed consists of two convolutional branches followed by a fully connected branch (structure inspired by [3]). The net can be sketched as in III-A, but considering two convolutional layers in the upper part instead of the fully connected ones.

Each upper branch has the following structure:

- 3 convolutional blocks consisting of a **Conv3d** layer followed by a **BatchNorm3d** layer.
  We have chosen the size of the convolutional kernel equal to 5, the padding equal to $\frac{5-1}{2}$ (in order to perform *valid padding*) and stride equal to 1. The normalization layer has the effect of normalizing the input of each layer, using mean and variance of a batch.
- **LeakyReLU** as activation function, which has the same benefits of the already discussed **ELU**.
- A final pooling layer with **AvgPool3d**; pooling layers produce downsampling, that is reduction of the spatial size of the convolved feature. Our choice, *average pooling*, returns the average value of the portion of the convolved feature covered by the kernel, which has size 2.

The outputs from the two branches are then concatenated along the channel dimension, and the result is flattened; this is then processed by the final fully connected branch, including 5 blocks of the form **Linear** + **Activation** + **Dropout** ($p = 0.1$) and, when the output size is reduced to 16, 3 final **Linear** layers.

V. Results

In this section we report the results obtained for the FNN and the CNN. At first, we tried to use the same number of training and test points and epochs in order to make a suitable comparison, but we found out that the FNN results do not substantially improve with more than 7.000 dataset points.

First of all, we report two significant types of plot for the performance evaluation of the NN; in the *correlation plots* we compare, on the set of validation points, the output of the net with the true value from the RT simulation. Ideally, we would like the plot to be stretched on the diagonal, which corresponds to the case of exact prediction; the other lines reported are $y = x \pm \frac{0.68}{2}x$, and delimit the range in which we expect that half of the data fall (from the quantiles of the Standard Normal).

We also report the trend of the training and test loss with respect to the number of epochs. The model which shows the best performance is indicated with a small red cross.

In the same plot, on the right, we report the trend of the R^2 score.

It is defined as:

$$R^2 = 1 - \frac{\sum_{j=1}^{n}(A_j - P_j)^2}{\sum_{j=1}^{n}(A_j - \bar{A})^2}$$

where A_j is the ground truth, P_j is the predicted value and $\bar{A}$ is the mean of the ground truth.

A. Same dataset, different networks

1) Fully Connected Neural Network: from the analysis of the correlation plot, the area in which the prediction is worse corresponds to points with low ionisation rate.

Moreover, we found out that the results obtained were better if we removed the final sigmoid activation function.

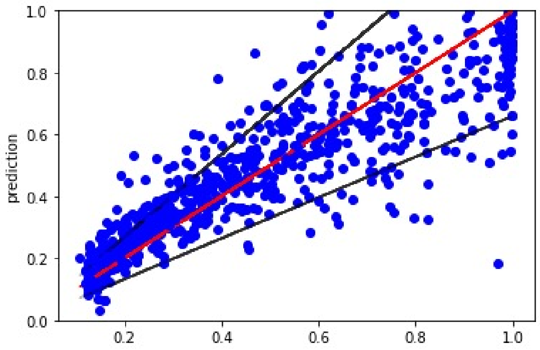


Fig. 2. Correlation plot: FNN with 6.300 training points, 700 validation points

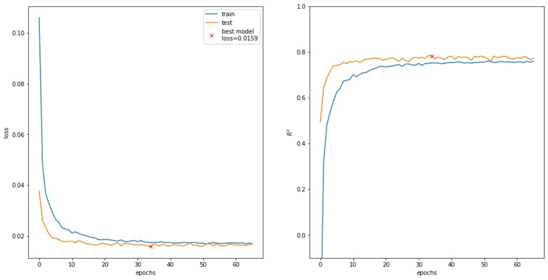


Fig. 3. Plot of loss trends (left) and R^2 score: FNN with 6.300 training points, 700 validation points

2) Convolutional Neural Network: we first made some unsuccessful attempts with radius of neighborhood equal to 24. In this case, we were forced to use only 3.000 input data due to memory limitations, and the results turned out to be critical (see Figure 4). We performed trials for decreasing values of neighborhood size, and we understood that, to prevent overfitting, increasing the input data amount is more effective than using a large neighborhood. Moreover, the *Mean Free Path* only sets a hard limit after which photons, of a source at the central pixel, are supposed to be completely depleted.

In reality, photons may stop before, and it actually depends on the cosmic time. The fact that we got the best result for a radius of 4 (see Figure 6) could mean also that this is the sphere of influence of the sources, at this epoch of reionisation.

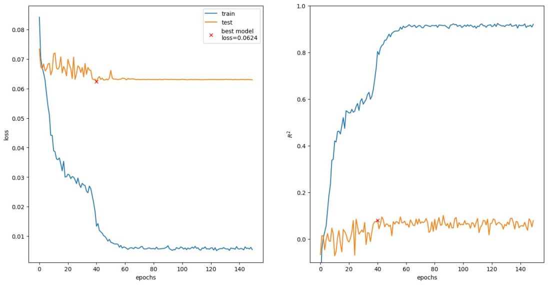


Fig. 4. Plot of loss trends (left) and R^2 score (right): CNN with neighborhoods of size 24, 2300 training points, 700 validation points. The excellent results for the training and the inadequate results for the test clearly suggest a phenomenon of overfitting.

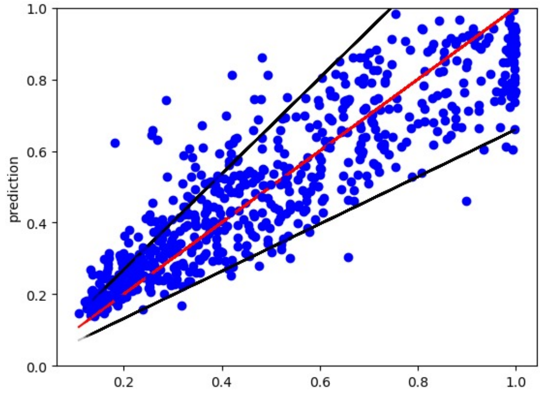


Fig. 5. Correlation plot: CNN with 69.300 training points, 700 validation points.

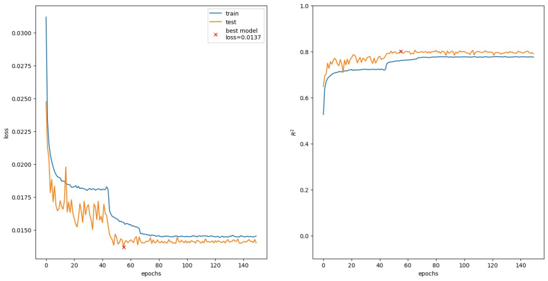


Fig. 6. Plot of loss trends (left) and R^2 score (right): CNN with 69.300 training points, 700 validation points.

B. Looking for the best combination

The model providing the best output, considering the R^2 score, is the FNN. However, we noticed from the correlation plots that the CNN performed better on low ionisation points, and that the expressive power of the CNN could even be enhanced more by increasing the number of training points.

VI. Conclusions

Using the largest amount of data possible for our computational resources (70.000), we succeeded in removing the overfitting of the initial trials with fewer data (3.000).

At that stage of the project, indeed, we suspected that the *overfitting* behaviour observed was probably due to the limited number of input data points, and the definitely better results obtained with 70.000 points confirmed our intuition that the previous models turned out to be tailor-made for our small sample, and failed in fitting the validation points well.

In the best CNN model, the R^2 score for the validation set is close to the training set's one, and it is ~ 0.7 (ideally, $R^2 = 1$ represents the best score achievable).

We think that it could be even increased, since, even in the case in which we exploit the largest number of data points, we still use only the $\sim 0.26\%$ of the availability.

VII. Further Developments

Another tool we could exploit to improve the performance is **data augmentation**: in our case, a strategy to increase the number of data available for training could be rotating each subvolume and this should be preferably done on the fly before training, without storing additional files.

However, we remark that this would significantly increase the computational time, and it has not been possible for this project due to the time limitation.

Another technique that could be adopted is a concatenation between a CNN and a FNN, as we have seen that the FNN provides better results and the CNN has the potentiality to perform even better, but is limited by the computational effort. The concatenation would exploit together the benefits of the two models.

This project, after having refined the working CNN model for a given redshift z, could be the starting point for the addition of the information provided by different times, that is, in our case, different z. In this framework, we leave the door open to the development of a *Long short-term memory* (LSTM) Neural Network, which can process not only single data points, but also entire sequences of data.

Acknowledgements

The authors thank their mentor Dr. Michele Bianco for the passion he has transmitted and the support that he has offered us during these five weeks of project work.

References

[1] T. R. Choudhury and A. Ferrara, "Physics of cosmic reionization," 2006.

[2] M. Bianco, S. K. Giri, I. T. Iliev, and G. Mellema, "Deep learning approach for identification of h_{ii} regions during reionization in 21cm observations," *Monthly Notices of the Royal Astronomical Society*, vol. 505, no. 3, p. 3982–3997, May 2021. [Online]. Available: http://dx.doi.org/10.1093/mnras/stab1518

[3] D. Prelogović, A. Mesinger, S. Murray, G. Fiameni, and N. Gillet, "Machine learning astrophysics from 21cm lightcones: impact of network architectures and signal contamination," *Monthly Notices of the Royal Astronomical Society*, vol. 509, no. 3, p. 3852–3867, Nov 2021. [Online]. Available: http://dx.doi.org/10.1093/mnras/stab3215