Modelling of the neutral hydrogen in galaxies with a Fully-Connected Neural Network

Brioschi Riccardo, D'Angeli Gabriele, Di Gennaro Federico
Tutor: Michele Bianco
Laboratory of Astrophysics (LASTRO), EPFL, Switzerland

***Abstract*—In this paper we approached the problem of finding a functional representation of the mass of hydrogen in galaxies with ML techniques. The main goal of the project was showing that current models used to describe the mass of hydrogen are too simplistic. Our solution uses a Fully-Connected Neural Network, that points out the complex relation among the mass of hydrogen and other physical, astrophysical and cosmological features.**

I. Introduction

In cosmology, Baryon Acoustic Oscillations (BAO) are fluctuations in the density of the visible baryonic matter (baryons) of the universe that was imprinted by density waves as an effect of the Big Bang. One of the best way to observe the imprinting of BAO on the baryonic density distribution of our Universe is looking at cosmic neutral hydrogen (HI).

The Hydrogen Intensity and Real-time Analysis eXperiment (HIRAX)[1] is a radio array telescope that will directly detect the neutral hydrogen mentioned above in the early stage of galaxy cluster formation; this means that we are observing the Universe 10 to 7 billion years back in the past. During this period, most neutral hydrogen is expected to be present only inside galaxies [2]. Therefore, the relationship between the amount of neutral hydrogen and the hosting galaxy mass is essential for this experiment and it must be better understood before being applied to HIRAX observations.

In order to depict the hydro-dynamic evolution of the galaxy's gas content, we have used accurate numerical simulations provided by CAMELS (Cosmology and Astrophysical with MachinE Learning Simulations)[5] project that recreate neutral hydrogen trapped inside galaxies. The hydro-dynamic simulations show a M_{HI} vs M_{halo} relation more complex than the standard one-to-one models used so far in cosmology[3][4] (Fig 1). For this reason, our goal is to train a Fully-Connected Neural Network that, given cosmological, astrophysical and galaxies' properties within a halo, is able to catch the scatter relationship between M_{HI} and M_{halo} .

To do so, the following N_{feat} features have been used:

- M_{halo} : mass of the halo (in solar mass units)
- V_{Halo} : euclidean norm of the velocity of the halo
- N_{subs}: number of galaxies within each halo
- M_{BH} : cumulative black hole mass within the halo (in solar mass units)
- $\dot{M}_{BH}$: black hole accretion mass (in solar mass per year units)
- P_{halo} : position of the halo with respect to the center of the simulation

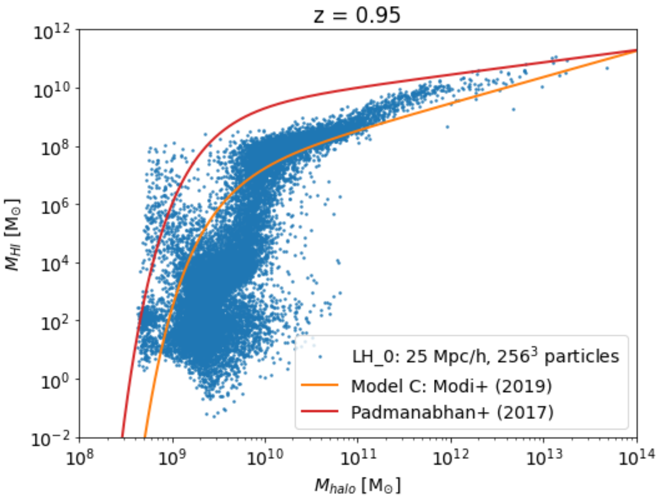

Fig. 1: *Scatter plot of neutral hydrogen and halo mass.*

- $Z_{gas,i}$, with $i = 0, 1, 2$ for hydrogen, helium and heavier elements respectively: fraction of metals in the gas within the halo
- $Z_{star,i}$, with $i = 0, 1, 2$ for hydrogen, helium and heavier elements respectively: fraction of metals in the stars within the halo
- SFR : star formation rate (in solar mass per year units)
- F_{AGN} : the average radiation flux within the halo
- T : temperature of the gas in the halo
- ρ : the gas density in the halo
- M_{HI} : neutral hydrogen mass within the halo (in solar mass units) (target)
- z : redshift of the halo

Data are divided depending on both the redshift parameter z and the astrophysical and cosmology constants used to run the simulations. For each fixed set of such cosmological and astrophysical costants (LH$_i$ folder), there are 16 simulations, corresponding to different values of the redshift z.

II. Exploratory Analysis

Our first goal was becoming familiar with the features in order to understand whether they were appropriate to train a well-suited Neural Network. In this section, we explain the exploratory analysis of the dataset. First, we remark that from literature[3][4] it is known that both z and M_{halo} are essential features to predict M_{HI}.

We decided to start using data from a single simulation, namely fixing the simulation parameter (LH) and the redshift z randomly (LH_0 and $z = 2.630$, hereinafter). We observed that $Z_{gas,i}$ and $Z_{star,i}$ are not informative features, since they have the same values for almost all the observations. Therefore, we decided to not use them in our work. Moreover, since P_{halo} is related to the geometry of the simulation and does not have an actual physical meaning, we decided to drop it. We also noticed that M_{halo}, V_{halo} and T have very skewed distributions. For this reason, we applied a logarithmic transformation to reduce their skewness ($x_{new} = log(1 + x_{old})$). This way of handling data will be performed even when training the Neural Network in Section IV.

In order to quantify the impact of the remaining features on the target, we decided to use the K-Means Clustering Algorithm. Our aim was to investigate if these covariates introduced a meaningful clustering among the observations. Since the algorithm is based on the notion of distance, we standardized the features to avoid having this unsupervised algorithm overparameterized with respect to the covariates with the largest values. We used the so called *"Elbow method"* (Fig 2) to get the best value of K (i.e. the number of clusters). Setting $K = 5$, we obtained the clustering depicted in Fig 2.

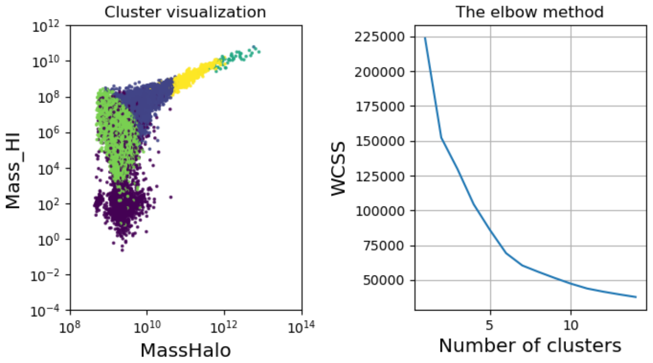

Fig. 2: *Clusters visualization and elbow method.*

The cluster visualization plot points out that this unsupervised algorithm performs quite well on this task: data-points are well divided by horizontal lines depending on specific ranges of M_{HI}. However, there are still some problems that might be related to the lack of spatial correlation in our data, an important assumption when using K-Means Clustering. The analysis can be generalized for different combinations of simulation parameter and z, obtaining a similar behavior. Therefore we concluded that the analysed features revealed to be good explanatory variables.

III. Data Processing and Architecture

A. Pre-processing

After the exploratory analysis of the features described in Section II, data have been processed to be used to train a Fully-Connected Neural Network as follows:

1) Log-transform of target: Even if we aim at approximating M_{HI} in each halo, we cannot expect the network to accurately learn these quantities, which are characterized by a considerable order of magnitude. Indeed, large numbers might cause problems in the learning process when computing the back-propagation algorithm. Therefore, we decided to compute the logarithmic transformation of the target, defining $y_{new} = log(1 + y_{old})$. By doing so, we aim to predict the order of magnitude of M_{HI}.

2) Masking: In the context of the exploratory analysis, we noticed that most of the observations had three features ($M_{BH}, \dot{M}_{BH}$ and SFR) with almost 95% of the values being equal to zero. These data-points correspond to halos with a $Mass_{Halo}$ value lower than 10^{10} solar mass units. So, we decided to divide the problem in two cases, training a different NN for each framework. In the first case (*masking case*) we only kept halos having the three features all different from zero; in the second case (*NO masking case*), we kept the whole dataset without filtering any data. The choice of considering these two different cases was supported also by a physical reason: the radio array telescope of the HIRAX, indeed, will mainly detect cosmic neutral hydrogen coming from high mass halos (i.e. halos with mass higher than 10^{10}, corresponding to our masking case). For this reason it is worth to separately and carefully investigate the M_{HI} vs M_{halo} relation for the specific case of *high-mass halos*.

3) Scenarios: As explained in the previous sections, we had to deal with simulated data in four different cases, respectively specifying or not the chosen LH folder and the used redshift parameter z. Here we will briefly discuss the main differences in each of the four scenarios:

- *Scenario 1* (LH_i fixed, z_j fixed): data obtained from a single simulation;
- *Scenario 2* (all LH_i, z_j fixed): data coming from simulations having same redshift parameter but different cosmological and astrophysical constants;
- *Scenario 3* (LH_i fixed, all z_j): data coming from simulations having same cosmological and astrophysical constants but different redshift parameter;
- *Scenario 4* (all LH_i, all z_j): the whole data-set. Both constants and parameter are important features to be taken into consideration.

For the last three scenarios we computed both the masking and no masking procedures.

4) Standardization: A good practice in ML is to standardize the data-set to reduce the possibility of overflow. We normalized our data using the mean μ and the std. deviation σ $\left(X_i' = \frac{X_i - \mu}{\sigma}\right)$.

B. Network Architecture

It is important to emphasize that we developed two different networks conditionally on the procedure we utilized (*masking* or *no masking case*). Even though initially the two models shared the same architecture, they ended up with different sets of hyperparameters. As far as the activation function is concerned, we considered ReLU and LeakyReLU (with a slope α of 0.1). The number and the size of hidden layers as well as the specific activation function and the possibility of Dropout in every layer were chosen via cross-validation.

IV. Training

A. *First architecture on scenario 1*

We started working on the first scenario after choosing a random simulation (e.g. LH_0 and $z = 0.95$, hereinafter). We trained a neural network with 4 hidden layers, 16 nodes in the initial layer and ReLU as activation function. The number of nodes in the hidden layers follows the power of two, increasing from one layer to the following one. Since, in this scenario, we had at our disposal just few datapoints (≈ 600), we considered all the observations without applying the masking procedure. As the correlation plot (Fig. 3) suggests, this model was not powerful enough to capture the hidden relation between M_{HI} and the input features.

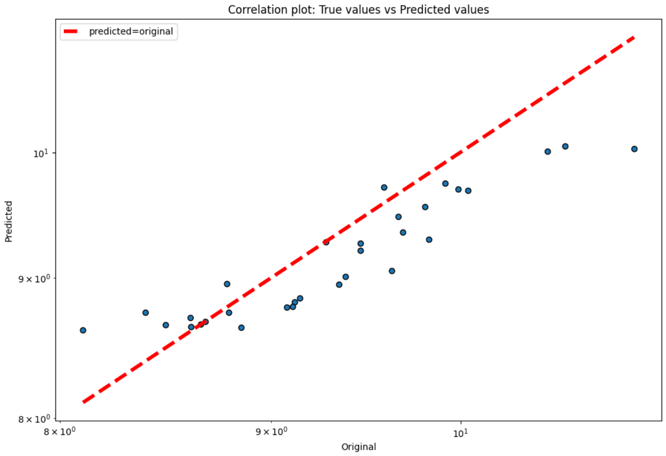

Fig. 3: *Correlation plot in log scale, using target values and predictions transformed as mentioned in II.*

B. *Optimization of hyperparameters and general results*

From the early beginning of the NN's training, we observed that different architectures led to different results in terms of test loss (MSE loss) and R^2 score (additional metric used to evaluate the performance of the models). To increase the predictive power of our network and to better capture the relationship between M_{HI} and M_{halo}, we determined the best set of hyperparameters using `Talos`[1] library.

Here are described the different combinations of hyperparameters we chose to test. It is necessary to note that, given $layer_{size}$ = K, the network will be composed by N layers (including both hidden and output layer) of size $(N_{feat}, K) \rightarrow ... \rightarrow (2^{i-1}K, 2^iK) \rightarrow ... \rightarrow (2^{N-1}K, 1)$

- N = number of layers: {3,4}
- K = layer size: {16,32,64}
- Activation function: {ReLU,LeakyReLU(α=0.1)}
- lr=learning rate: {10^{-2}, 10^{-3}}
- D = Dropout rate: {0.05, 0.1}

Regarding the backward step, we decided to compute Stochastic Gradient Descent. For each combination of these hyperparameters we trained a model for 100 epochs. Finally, we chose the best model comparing validation loss values. To reduce the computational complexity of this optimization process, instead of directly working with the whole data-set (*scenario 4*), we separately computed the same optimization algorithm for both *scenarios 2* and *3*. We repeated this approach twice, accordingly to the two procedures explained above (masking and no masking).

The following tables show the best models for both scenarios:

Lab	N	K	Activation function	lr	D	val loss
A	**4**	**16**	**ReLU**	**0.01**	**0.05**	**0.41**
B	4	32	ReLU	0.01	0.10	0.58
C	4	32	LeakyReLU	0.01	0.10	0.65
A'	4	32	LeakyReLU	0.01	0.05	5.58
B'	**4**	**32**	**LeakyReLU**	**0.01**	**0.10**	**5.60**
C'	3	32	ReLU	0.01	0.10	5.61

Table I: *Top 3 models in case of masking for Scenario 3 above the double line. Top 3 models in case of NO masking for Scenario 3 below the double line.*

Lab	N	K	Activation function	lr	D	val loss
E	4	16	LeakyReLU	0.01	0.10	0.34
A	**4**	**16**	**ReLU**	**0.01**	**0.05**	**0.60**
F	3	32	ReLU	0.01	0.05	0.69
D'	4	16	ReLU	0.01	0.10	5.54
E'	3	32	LeakyReLU	0.01	0.05	5.55
B'	**4**	**32**	**LeakyReLU**	**0.01**	**0.10**	**5.56**

Table II: *Top 3 models in case of masking for Scenario 2 above the double line. Top 3 models in case of NO masking for Scenario 2 below the double line.*

In order to find the best model to use when working with the whole data-set (we did the same procedure for both masking and not masking case separately), we cross checked the rankings in Table I and Table II (considering the results obtained for the procedure we decided to use) and we chose the 2 models appearing in the highest positions in both rankings.

1) Scenario 4 for masking case: The chosen model was the one having label *A* in Table I and Table II (upper part of them). With a final R^2 score of 0.848 and a test loss of 0.03 (after 600 epochs of training), this model correctly captures the M_{HI} vs M_{halo} relation for halos having large M_{halo} values as shown in Fig. 4.

2) Scenario 4 in NO masking case: The chosen model was the one having label *B'* in Table I and Table II (bottom part of them). With a final R^2 score of 0.882 and a test loss of 0.59 (after 600 epochs of training), this model correctly captures the M_{HI} vs M_{halo} relation for the majority of halos generated by the simulations, without restrictions on M_{halo} values (Fig. 5).

Given the fact that the correlation plot for this case was not as easily interpretable as the one in Fig. 3 (too many

[1] https://github.com/autonomio/talos

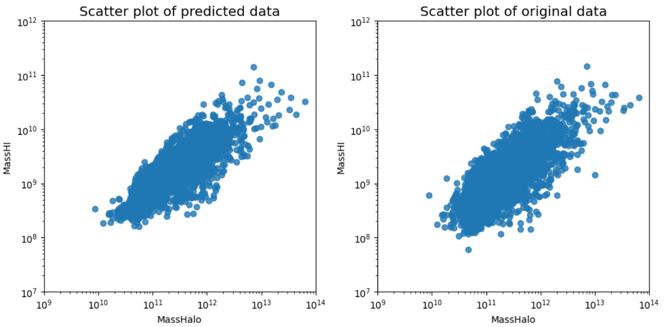

Fig. 4: *True cloud of points vs predicted cloud of points. The final network seems to reproduce the general structure of the cloud accurately.*

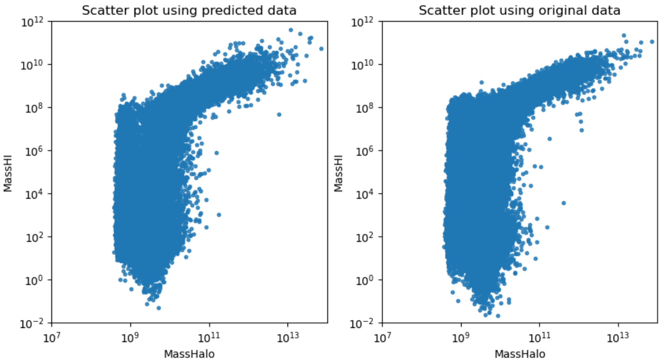

Fig. 5: *True cloud of points vs predicted cloud of points. The final network seems to reproduce the general structure of the cloud accurately.*

datapoints overlapping in different regions), we decided to visualize the goodness of fit of the final model (label *B'*) by using a correlation plot that captured the density of points in each region (Fig. 6). It is important to notice that, while comparisons between clouds of points (as in Fig. 4 and Fig. 5) give us a general overview of the performance of the networks, correlation plot offers an informative account of the goodness of fit for each datapoint.

V. Conclusions

Considering the results described in Section IV, we can claim that the main goal of the task has been accomplished. The Fully-Connected Neural Network with the best hyper-parameters for the general case (*Scenario 4, NO masking*) is indeed able to capture the scatter relationship between M_{HI} and M_{halo} shown in Fig. 1. This suggests that the underlying relationship between M_{HI} and M_{halo} is probably more complex than the one described by the models proposed so far [3][4]. An additional improvement could be obtained by adding features more relevant to this physical problem: features $F_{AGN}, M_{BH}, \dot{M}_{BH}, SFR$ contained around 95% of zeros, therefore not allowing the network to use them in order to discriminate the actual value of the target. In addition, this might be the reason why the loss (of the test set) moved from 0.03 when performing the *"masking"* procedure to a value of 0.59 in the *"no masking"* case. Another detail that might positively affect the quality of the research is about simulating and handling data. With the simulations provided in [5], it is not possible to keep track of halos across different redshifts z. This leads to the problem of not being sure of having train and test sets completely independent (the same halo could appear in the train and test sets with different redshift values). If it was possible to keep track of the temporal evolution of each halo, then the problem could be further investigated by using a more appropriate architecture, such as a Long Short Term Memory network.

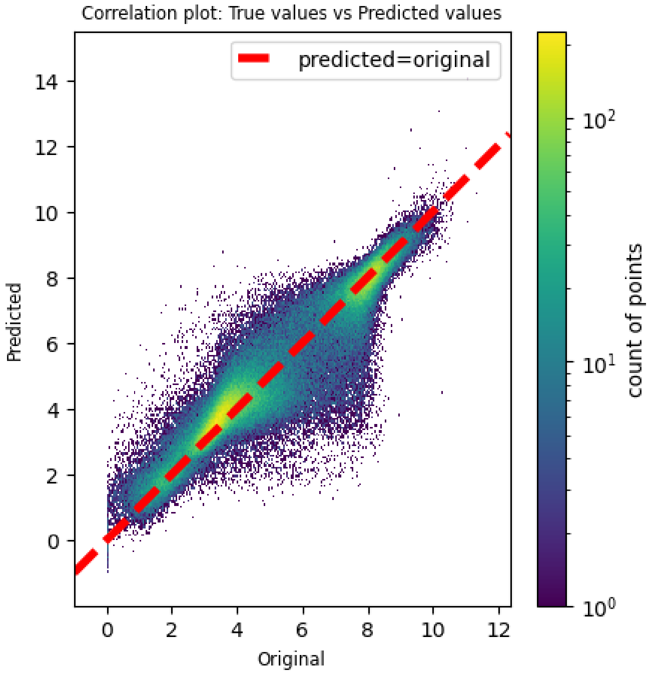

Fig. 6: *Correlation plot in log scale, using target values and predictions transformed as mentioned in II*

VI. Acknowledgemts

We would like to thank our tutor Dr. Michele Bianco for the support provided during the project and for the opportunity he gave us to work on this interesting topic in collaboration with LASTRO.

References

[1] HIRAX: A Probe of Dark Energy and Radio Transients. L.B. Newburgh (Toronto U.). *K. Bandura, M.A. Bucher (APC, Paris), T.-C. Chang (Taiwan, Natl. Taiwan U.), H.C. Chiang (KwaZulu Natal U.) et al.*.

[2] Modeling the neutral hydrogen distribution in the post-reionization Universe: intensity mapping. *Francisco Villaescusa-Navarro, Matteo Viel, Kanan K. Datta, T. Roy Choudhur*.

[3] Intensity mapping with neutral hydrogen and the Hidden Valley simulations. *Chirag Modi, Emanuele Castorina, Yu Feng, Martin White*.

[4] A halo model for cosmological neutral hydrogen: abundances and clustering. *Hamsa Padmanabhan, Alexandre Refregier, Adam Amara*.

[5] The CAMELS project: Cosmology and Astrophysics with MachinE Learning Simulations. *Francisco Villaescusa-Navarro, Daniel Anglés-Alcázar, Shy Genel, et al.*