ML4Science:
Week 1 Meeting

Calafà, M., Mescolini, G., Motta, P.

École Polytechnique Fédérale de Lausanne (EPFL)

Machine Learning Project 2

23 Nov 2021

Tutor: Dr. Michele Bianco

Outlines

1 Convolutionary Neural Networks: what do we know?
- Limits of FCNNs
- Structure of a CNN
- The advantages
- Some useful tools for the project

2 Organization of the work

3 Data Preprocessing
- Unit conversion
- Normalization

4 Questions

CNNs

Limits of FCNNs

Two critical issues of Fully Connected Neural Networks are:

- The number of parameters is extremely high ($\mathcal{O}(K^2L)$ for a NN with L hidden layers and K nodes per each layer).
- The information provided by the spatial dependencies is lost, since each node is treated without caring about its position.

A possible solution

These two issues can be solved with a Convolutionary Neural Network, which includes **convolutional layers** and **pooling layers** followed by a Fully Connected Neural Network.

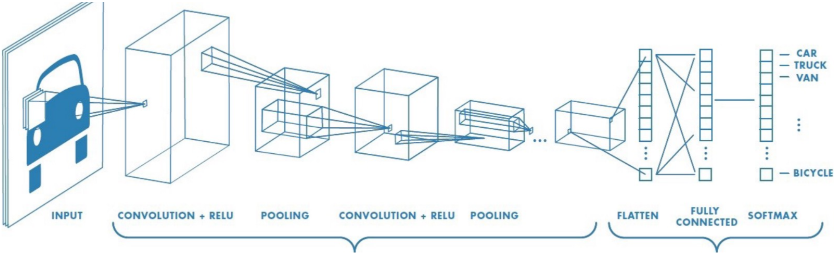


Figure: An example of CNN

Convolutional Layers

A convolutional layer has the effect of applying a local filter to the input, with a convolution operation; for example, the n-th component of $x^{(1)}$ is computed as:

$$x^{(1)}[n] = \sum_{k=0}^{K} f[k] x^{(0)}[n-k]$$

$f[n]$ is often a **local** filter, in the sense that $f[k] = 0$ for $|k| \geq \tilde{K}$. This applies in the case where x is a matrix, as well (in this case, f is clearly a matrix).

Convolutional Layers

It is possibile to apply multiple filters to the input, obtaining several outputs.
Those outputs are called **channels**, and the deeper we go into the network, the more channels we get, as it can be seen in Figure 1.

Pooling Layers

This kind of layers has the effect to perform **downsampling**, i.e., reduction of the spatial size of the convolved feature. In the lectures, we have seen the:

- Max Pooling
- Average Pooling

which return, respectively, the maximum value and the average value of the portion of feature covered by the kernel.

The advantages

- High number of parameters
  $\Longrightarrow$ Since the filter is local and we use the same filter at every position, the NN structure is sparse and local with few parameters.
- Missing of spatial dependencies
  $\Longrightarrow$ When a node undergoes the effect of the filter, its neighbours are involved.

Useful Techniques

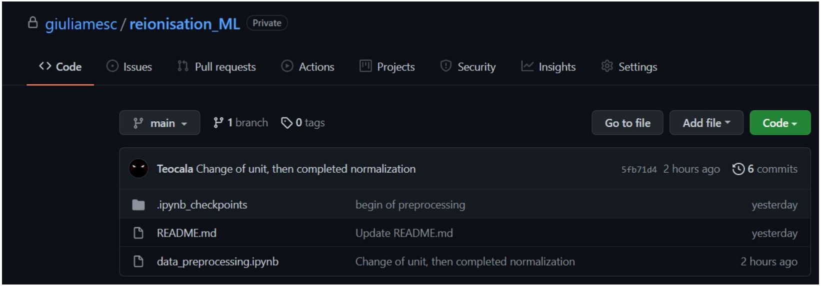
During the lectures, some strategies have been presented; we tried to reflect on their impact on this project.

- **Regularization**: may be useful if we have troubles with overfitting.
- **Data Augmentation**: does not seem a viable option to us, also because our problem is not a classification problem. However, rotations may be useful to reinforce isotropy.
- **Dropout**: may be useful.
- **Batch Normalization**: we have encountered the necessity to normalize our inputs; we have used a coarse method for the input data (see later), but we may refine it and adopt for every layer.

Organization

GitHub Repository

We are currently working on a GitHub private repository.



Up to now, we have worked with *Jupyter Notebooks*.

Preprocessing

Unit conversion

We noted that n_{img} and n_{src} are computed adopting two different units (the former is a density in the CGS unit system, the latter is instead a quantity per comoving mega-parsec).
Then, it is reasonable to convert them in the same unit system.

$$[n_{igm}] = \frac{g}{cm^3} = 10^{15}\frac{g}{km^3} = 10^{15} \cdot 3.09^3 \cdot 10^{39}\frac{g}{pc^3} = 2.95 \cdot 10^{55}\frac{g}{pc^3}$$

Therefore, we performed the following redefinition of n_{igm}:

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

Normalization

Successively, we analyzed more in detail what happens in each direction; we first chose to observe the behaviour of the 2 features and the output on the three orthogonal axes cutting the cube at its center of gravity.

Normalization

It emerges that the scale of the two features n_{igm} and n_{src} is extremely different. Therefore, as anticipated, the preprocessing of the data must clearly include a normalization.
As there is no significant difference of scale among the various dimensions, a first attempt can be normalizing the whole features using the total mean μ and standard variation σ.

MEAN VALUE ALONG EACH DIRECTION

	x	y	z
n_igm	1.1564589789595674e+25	1.1130535012706073e+25	1.1797878933858357e+25
n_src	20918.686235631307	17146.738710606893	19699.80898203532
xi	0.7047237362224594	0.6585446459674628	0.7675350039049594

Normalization

MIN AND MAX VALUES AFTER NORMALIZATION

	x	y
n_igm	-2.168842398716399	15.120823603045926
n_src	-0.5434308937395814	156.6221826854396
xi	-2.523045976013896	0.9082910299731384

Questions

Questions

- The meaning of max_{mpf} is not completely clear. Does it represent a subvolume or a radius? Is the value the same for every redshift?
- Suggestions for additional pre-processing transformations?
- Do you prefer having access to the GitHub repo?