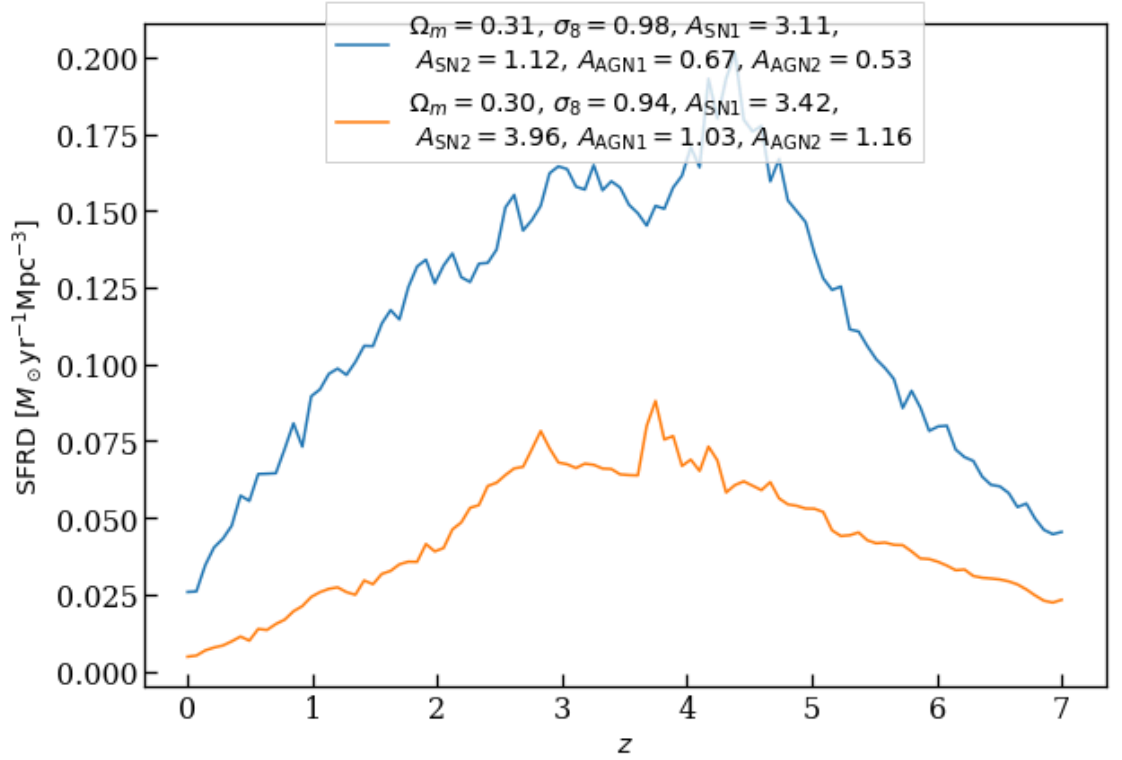


La star-formation rate density (SFRD), ovvero il tasso di formazione stellare per unità di volume, è una quantità che dipende, oltre che dall'età dell'Universo a cui la si misura (ovvero il redshift), dal modello cosmologico ed astrofisico:

$$\text{SFRD} = f(z, \vec{\theta})$$

dove  $\vec{\theta}$  è il vettore di parametri cosmologici ed astrofisici da cui dipende la grandezza. Dunque dalla misura della SFRD a diversi redshift è possibile, in linea di principio, ricavare i parametri cosmologici ed astrofisici da cui dipende.



Data la difficoltà di derivare un modello analitico per  $\text{SFRD} = f(z, \vec{\theta})$ , vogliamo approssimare con una rete neurale la funzione inversa  $f^{-1}(\text{SFRD}(z)) = \vec{\theta}$ , che mappa la star-formation rate density nel set di parametri da cui essa dipende.

A questo scopo useremo le misure di  $\text{SFRD}(z)$  ricavate da un set di 1000 simulazioni idrodinamiche, ognuna ottenuta con un diverso set dei parametri:

$$\vec{\theta} = \{\Omega_m, \sigma_8, A_{\text{SN1}}, A_{\text{SN2}}, A_{\text{AGN1}}, A_{\text{AGN2}}\},$$

dove i primi due parametri sono parametri cosmologici legati alla densità di materia e all'ampiezza delle fluttuazioni del campo di materia a  $z = 0$ , mentre i restati 4 parametri sono parametri astrofisici legati all'efficienza dei feedback energetici da esplosioni di supernovae e nuclei galattici attivi.

Il file `SFRH-IllustrisTNG.npy` contiene i dati della  $\text{SFRD}(z)$ , in unità di masse solari per anno su  $\text{Mpc}^3$ , misurata in ogni simulazione ai valori di redshift:

```
z = np.linspace(0.0, 7.0, 100) #redshifts of the SFRD
```

mentre il file `params-IllustrisTNG.txt` contiene i valori dei parametri  $\vec{\theta}$  utilizzati per creare le simulazioni.

- Selezionare 5 simulazioni, e produrre un grafico in cui vengono riportati, in funzione del redshift, le misure della SFRD(z) e i parametri corrispondenti (simile al grafico mostrato sopra).
- Definire la classe `make_dataset()` in cui i dati vengono preparati per essere analizzati dalla rete neurale. Suggestimenti: Normalizzare i valori dei parametri tra 0 e 1, e standardizzare i valori della SFRD; suddividere il data set secondo lo schema: 70% training, 15% validation, 15% test.
- Definire una rete neurale per predire, dati i valori di SFRD(z), i valori dei parametri  $\vec{\theta}$ , con un'architettura del tipo (Fully Connected Linear Network):

```
nn.Linear(input_size, hidden_size)
Activation # e.g. ReLU() or LeakyReLU()
nn.Linear(hidden_size, hidden_size)
Activation # e.g. ReLU() or LeakyReLU()
...
nn.Linear(hidden_size, output_size)
```

- Allenare la rete neurale variando il `learning_rate`, il numero di epoche, numero di hidden layers, ed il numero di neuroni per hidden layer (`hidden_size`). Quale set di hyper-parameters ottimizza la rete neurale? Argomentare il risultato.
- Fare un grafico con 6 pannelli, uno per ogni parametro predetto, che mostri il confronto tra i valori predetti dalla rete neurale con quelli veri per il test set. Commentare i risultati.
- Definire, ottimizzare ed allenare una rete neurale per predire, dati i valori di SFRD(z), i valori dei parametri  $\vec{\theta}$  ed il loro **errore**. Fare un grafico con 6 pannelli per confrontare le predizioni con il test set. Commentare i risultati.