deepsnap

- 官网地址： **https://snap.stanford.edu/deepsnap/**

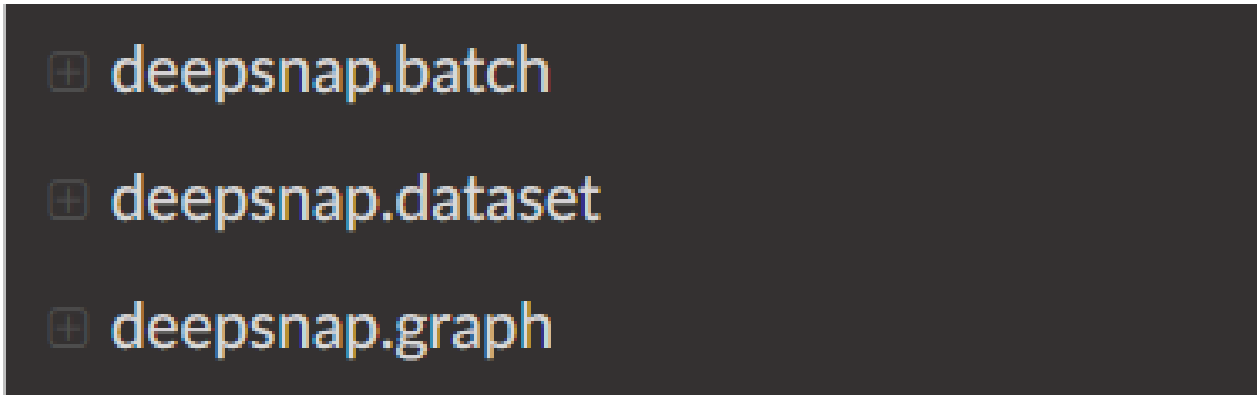
什么是 deep snap?

> DeepSNAP is a Python library to assist efficient deep learning on graphs. DeepSNAP features in its support for flexible graph manipulation, standard pipeline, heterogeneous graphs and simple API.

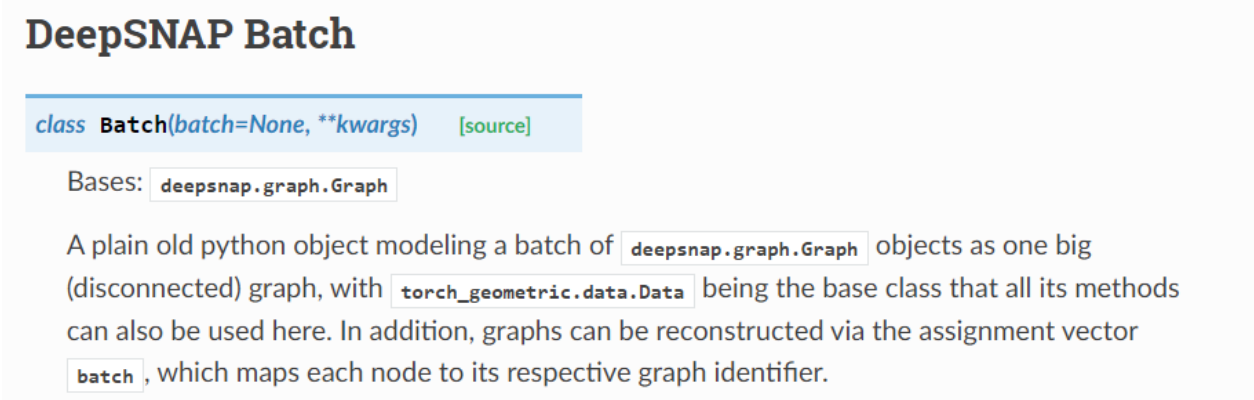
> 是一个Python函数库，用来帮助我们在图(graphs)上面进行**有效率**的深度学习，它支持图的操作，标准化流程，异构图和简单的API

类似的工具包有 PyTorch Geometric, DGL, GraphNets。

一些比较重要的 module!

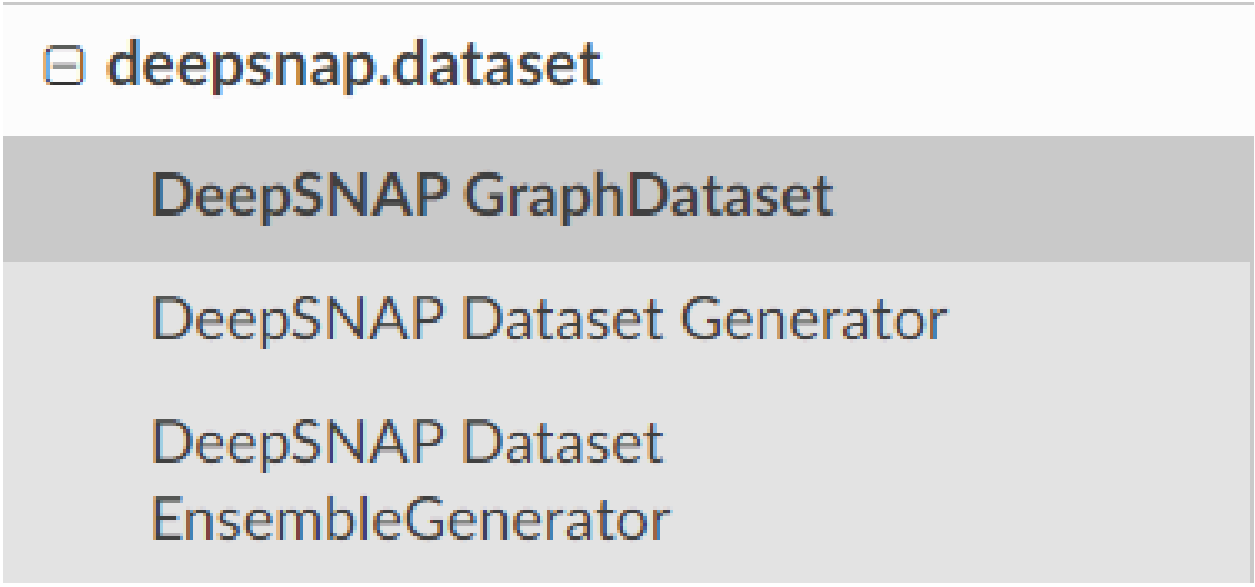


deepsnap.batch



大概意思应该是说 Batch 类可以将一批 Graph 集合成一个大的（不连接的）图，但是具体的用法 **官方文档** 上没讲.

deepsnap.dataset



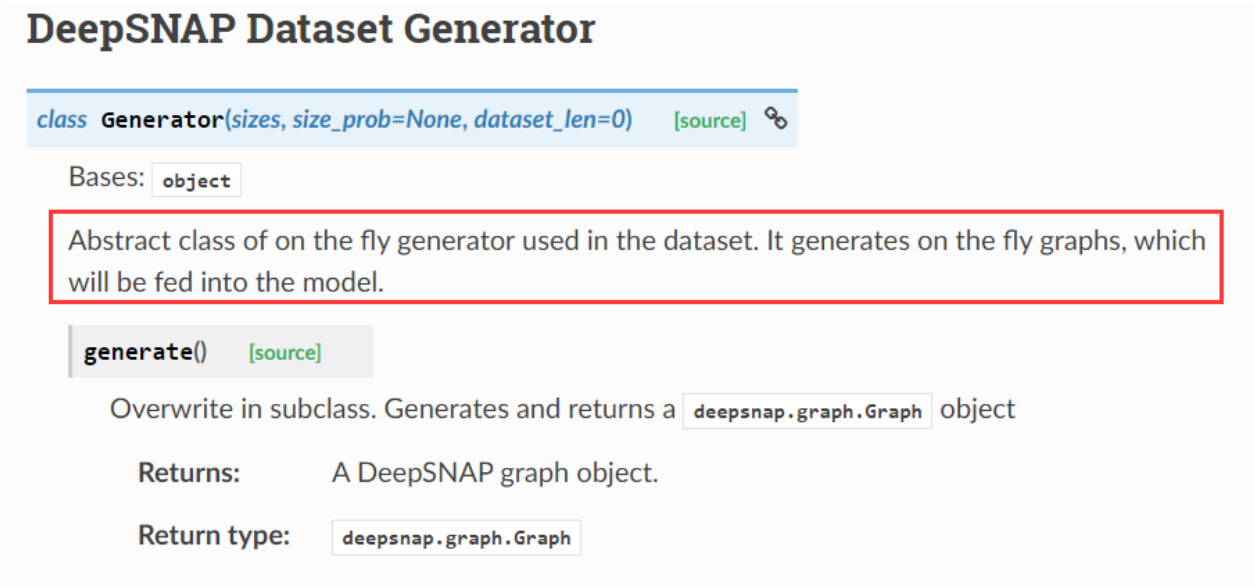
GraphDataset 类

```
classGraphDataset(graphs: Optional[List[deepsnap.graph.Graph]] = None,...)
```

- graphs (list, optional) – A list of deepsnap.graph.Graph.

用来存放图数据集列表

Generator 类



能够动态地生成各种图（ER graphs 等等，可以用 ERGenerator类 继承 Generator类 然后去重写 `generate()` 方法，`generate()` 方法中定义了图的产生方式

EnsembleGenerator 类

DeepSNAP Dataset EnsembleGenerator

class **EnsembleGenerator**(*generators, gen_prob=None, dataset_len=0*) [source]

Bases: `deepsnap.dataset.Generator`

generate(***kwargs*) [source]

Generate a list of graphs.

Returns: Generated a list of `deepsnap.graph.Graph` objects.

Return type: list

根据传进去的 generators 列表以 gen_prob 的概率生成 dataset_len 长度的图数据集。

重头戏来了：deepsnap.Graph 的格式

DeepSNAP Graph

class **Graph**(*G=None, netlib=None, **kwargs*) [source]

Bases: `object`

A plain python object modeling a single graph with various (optional) attributes.

Parameters:

- **G** (*Graph object, optional*) – The NetworkX or SnapX graph object which contains features and labels. If it is not specified, `Graph` will use the tensor backend.
- **netlib** (*types.ModuleType, optional*) – The graph backend module. Currently DeepSNAP supports the NetworkX and SnapX (for SnapX only the undirected homogeneous graph) as the graph backend. Default graph backend is the NetworkX.
- ****kwargs** (*optional*) – Keyworded argument list with keys such as `node_feature`, `node_label` and values that are corresponding attributes. The features are required for the tensor backend.

文档中说 `Graph.G` 是 **NetworkX or SnapX graph object**

在代码中输出

```
print(type(dataset[0].G))
```

```
output:
<class 'networkx.classes.graph.Graph'>
```

发现它确实是一个 networkx 包定义的 graph，于是我去找 networkx 包

那么要如何生成 network 的 graph 呢?

networkx官网 上有讲如何生成一张空的图，并往图中加节点和边，明天再看看

..