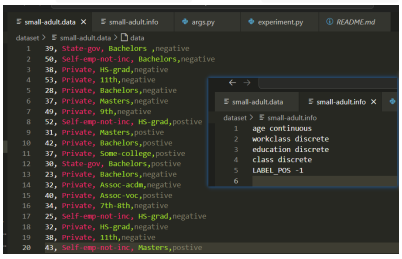
> **为了方便后面的学习，这里重新用了一个很小的数据集：**

- 数据集介绍：
  名字：`small-adult.data`。解释：`small-adult.info`。

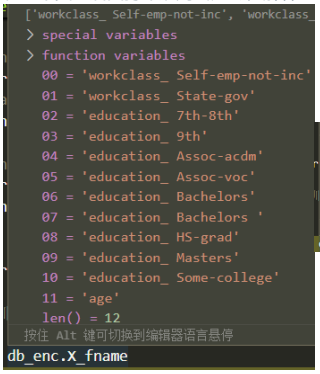


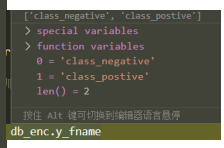
  蓝色框框里面是数据类型解释、介绍。
  也是最后会放到 提取的rrl 里面的关键物品。

讲解整个train的过程来理解网络结构.

(btw,现在所有的参数都在args.py里面default里面设置好了，每一次运行只需要直接运行程序即可)。

- X_df里存储所有特征。
  对离散特征处理之后，由(20,3)转换为(20,12)。12列里面最后一列是 连续特征。
  discrete_flen = 11; continuous_flen = 1;
  - 有两个变量后面打印规则会用到，解释一下：x_fname, y_fname存储每一列的实际意义。可以看下面的解释。

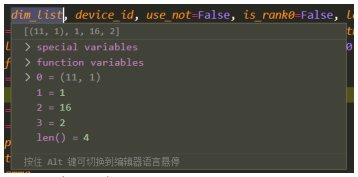




  - 网络结构初始化：

```
rrl = RRL(dim_list=[(discrete_flen, continuous_flen)] + list(map(int, args.structure.split('@'))) + [len(y_fname)],
          device_id=device_id,
          use_not=args.use_not,
          is_rank0=is_rank0,
          log_file=args.log,
          writer=writer,
          save_best=args.save_best,
          estimated_grad=args.estimated_grad,
          use_skip=args.skip,
          save_path=args.model,
          use_nlaf=args.nlaf,
          alpha=args.alpha,
          beta=args.beta,
          gamma=args.gamma,
          temperature=args.temp)
#使用多个参数初始化
```

    - 网络结构：**

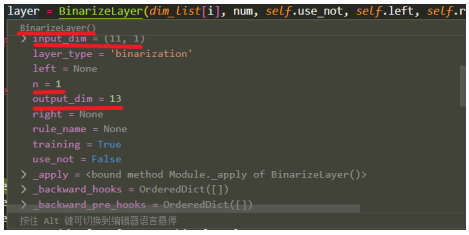


    第一个$(11,1)$表示11个离散特征、1个连续特征。第二个参数是二值化层次结点个数是1。
    第三个参数是unionlayer结点个数16。
    最后一个参数2是因为最后的分类的结果一共有两种。
  - 在rrl初始化里面，初始化整个网络结构的代码：

```
self.net = Net(dim_list, use_not=use_not, left=left, right=right, use_nlaf=use_nlaf, estimated_grad=estimated_grad, use_skip=use_skip, alpha=alpha
```

    这一步才是真正的网络结构、权重的初始化：

    - 二值化层次：
      一个结点，输入是11离散，1连续；最后输出维度是13.
      这里有一个问题，没有很明白，n = 1其实应该是说明有1个结点，但是最后有13个输出，感觉应该理解为这一个层次有13个结点会好理解一些。每一个结点处理一个特征值。



- 逻辑层初始化：

```
layer = UnionLayer(dim_list[i], num, use_nlaf=use_nlaf, estimated_grad=estimated_grad, use_not=layer_use_
layer_name = 'union{}'.format(i)
```

这个unionlayer其实是对 合取 析取 的包装，其实一个层次里面有两部分组成：

```
self.con_layer = OriginalConjunctionLayer(self.n, self.input_dim, use_not=use_not, estimated_grad=estimated_gr
self.dis_layer = OriginalDisjunctionLayer(self.n, self.input_dim, use_not=use_not, estimated_grad=estimated_gr
```

之后会进行两个层次的初始化。

  - conj层次初始化之后结构：

  - disj层次结构同上。

两个层次连接起来就是整个 UnionLayer 的结构：

有16个结点，(其实是16个做conj，还有16个做disj)
输出维度是32。每一个结点进行一个输出。
输入维度是13。
对于conj层次，权重参数的尺寸大小应该是$(13, 16)$。
disj也有一个相同的**尺寸大小**的权重参数。

- LRLayer的初始化：

```
elif i == len(dim_list) - 1:
    layer = LRLayer(dim_list[i], num)
    layer_name = 'lr{}'.format(i)
```

一个全连接层，结构比较简单。
输入维度是32，输出维度是2。
输入维度：上一层的结点处理完之后，产生了32个输出。所以这一层次就是32个输入。
输出维度：因为最后的分类一共有两个种类，就是2个结点。

- 在每一层 初始化 之后，都会存储这个层次的name和参数信息：

```
self.add_module(layer_name, layer)
self.layer_list.append(layer)
```