示例：

输入数据：

|  |
| --- |
| 0 1 1 #学生0做第1题，作对了  0 3 1 #学生0做第3题，作对了  1 3 0 #学生1做第3题，作错了  1 3 0 |

这里令batch\_size=3，也就是三个学生一个batch。

输出数据:

加载一个batch的数据为字典，其中字段有：

**input\_x：**三层的列表，其中每个列表为学生的做题信息，里面的每个子列表是题目的作答情况，每题都用一个onehot编码的向量表示。例如input\_x shape: (32, 153, 248)表示batch\_size=32，把32个学生的数据作为一批。153表示每个学生的做题序列的长度为153，不够这个长度的用padding来补充。248表示总共有124道题，长度为248的数组的前124位是作对题的对应编号，后124位是做错题的对应编号。例如共有3题（0，1，2号），算上padding的-1号题，总共有4题，学生作对了第一题就可以表示为[0,1,0,0,0,0,0,0]，做错了第一题表示为[0,0,0,0,0,1,0,0]。

**target\_id：**两层列表，每个列表是一个学生的预测数据，每个元素是要预测的学生的做题的编号，取每条数据序列的后len(i)-1部分。比如完整的做题的题目编号序列是[0,1,1,2]，那么模型输入的信息的题号是[0,1,1]，输出的信息的对应题号是[1,1,2]。这里的id数据不用onehot编码。

|  |
| --- |
| Output 1 1 2  Rnn->rnn->rnn  input 0 1 1 |

**target\_correctness**：与target\_id同样的shape，表示对应的预测的学生的做题的正确与否。

**seq\_len**：一层的列表。表示每条序列的实际长度。例如上面padding过后每条数据的长度都为153，第一个学生只做了7道题，那么seq\_len第一个元素=7.

**max\_len**：当前序列最大长度。不同的batch可能有不同的最大长度，每个batch里面找出最长的做题序列，然后把同一个batch里面的其他数据都padding为这么长。

前四个字段的值用numpy的ndarray保存。第五个字段用np.int32保存。

输出数据用yield生成器，每次用next()来调用。

可参考dtk.ipynb中的DataGenerator代码块