本节讲解模型的加密和分段：

/\*

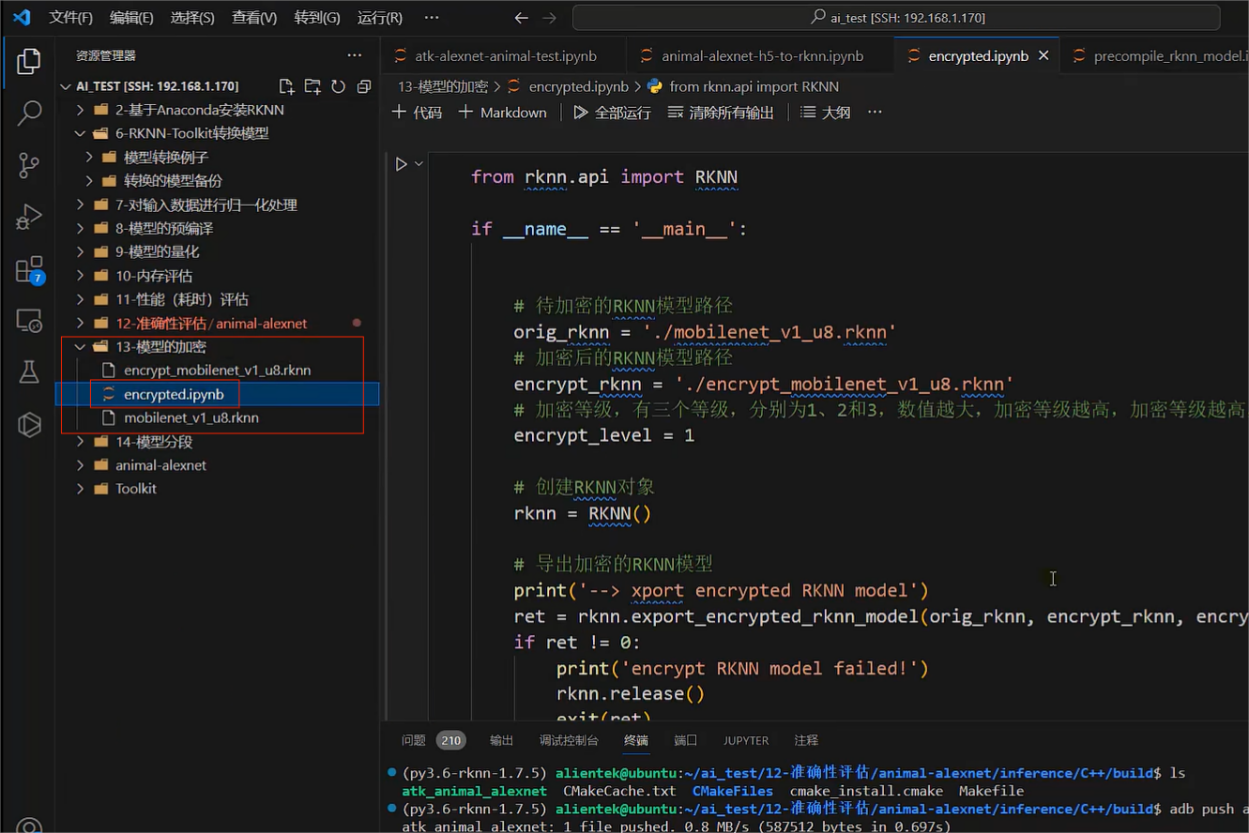


如果我们不想公开RKNN模型的结构，我们可以使用RKNN Toolkit里的模型加密功能对RKNN模型进行加密操作。加密后的RKNN模型在Netron这样的可视化工具下是无法查看其模型的结构的。加密后的RKNN模型在板端进行推理的时候，是需要由RKNPU完成模型的解密操作的。

\*/

/\*

模型加密的案例：



RKNN模型加密的api有3个参数：

orig\_rknn表示待加密的RKNN模型的路径

Encrypt\_rknn表示加密后的RKNN模型保存的路径

Encrypt\_lebel表示加密等级。有1，2，3共3个等级。数字越大等级越高，解密越耗时。

运行：

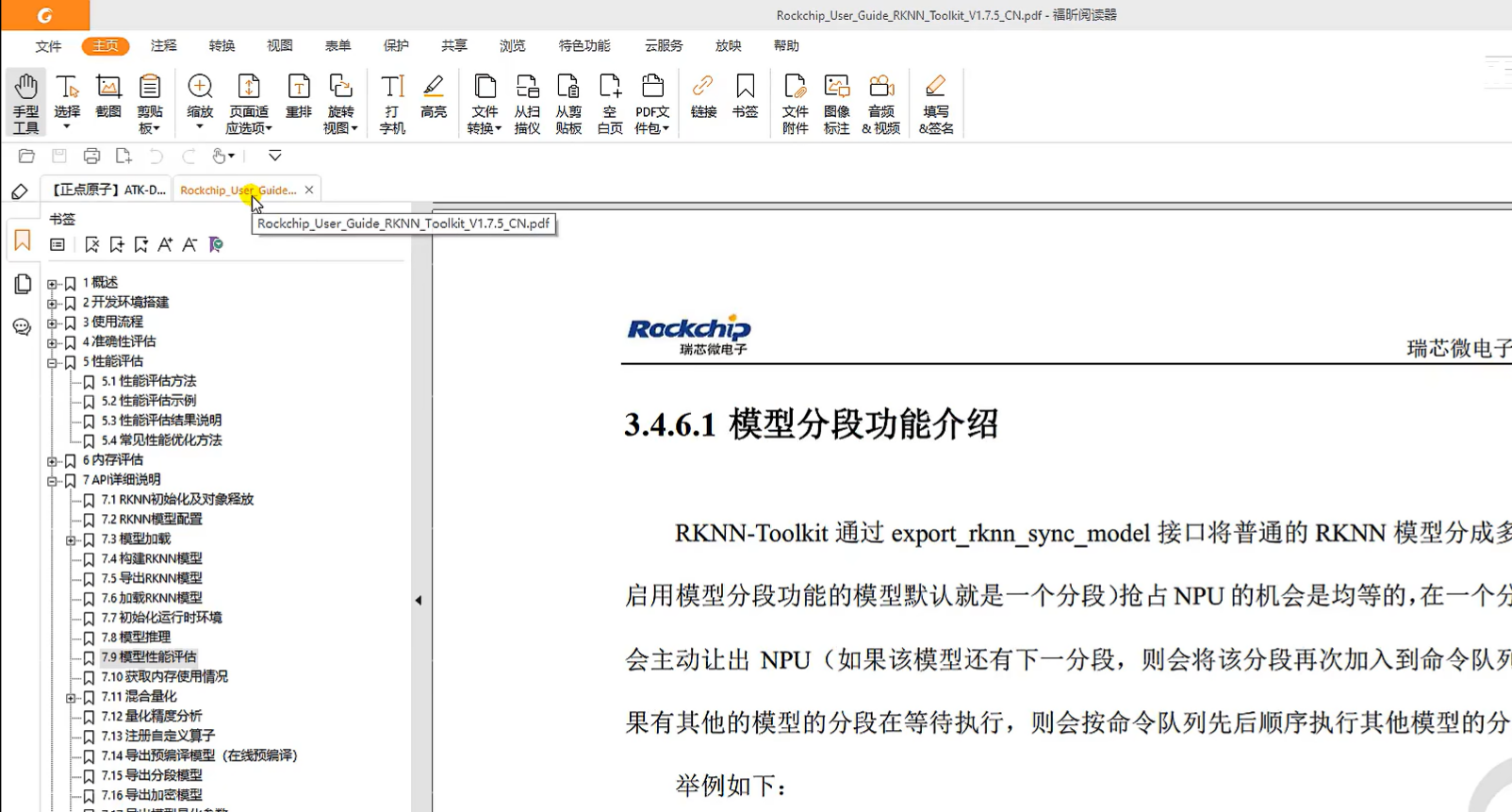
选择内核，然后选择前面在Ubuntu下搭建的虚拟环境。然后直接运行即可。这里不需要开发板通过ADB连接。运行后就对RKNN模型实现加密操作。

\*/

/\*

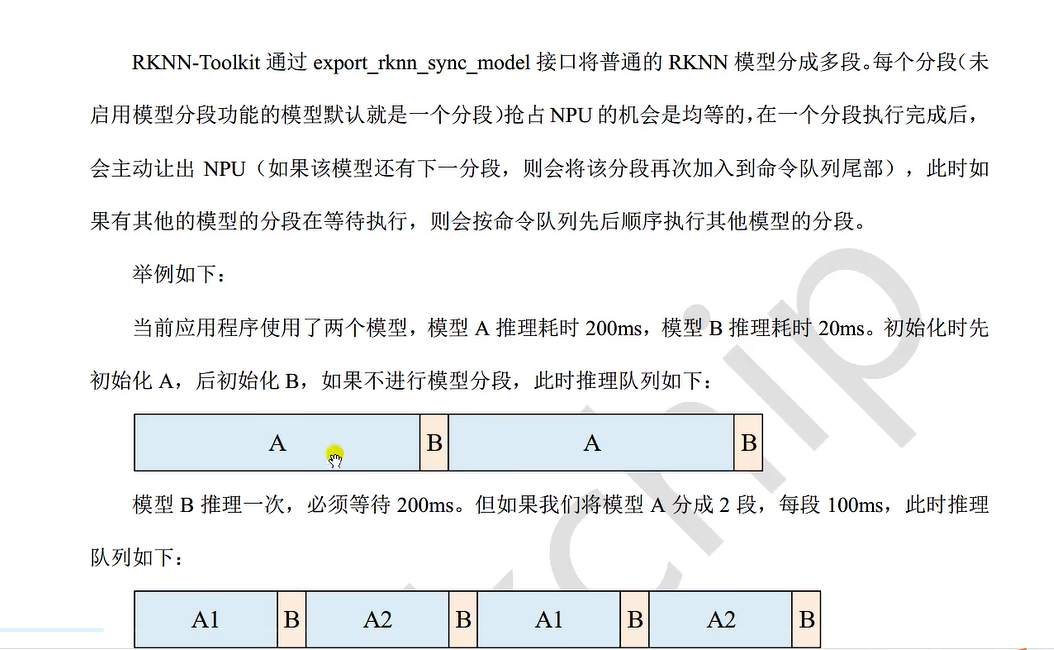
RKNN Toolkit里的模型分段功能：

模型的分段见RKNN Toolkit的手册介绍:



/\*

如果一个应用程序使用了两个/多个模型，如：模型A和模型B：



模型A推理耗时200ms，模型B推理耗时20ms，

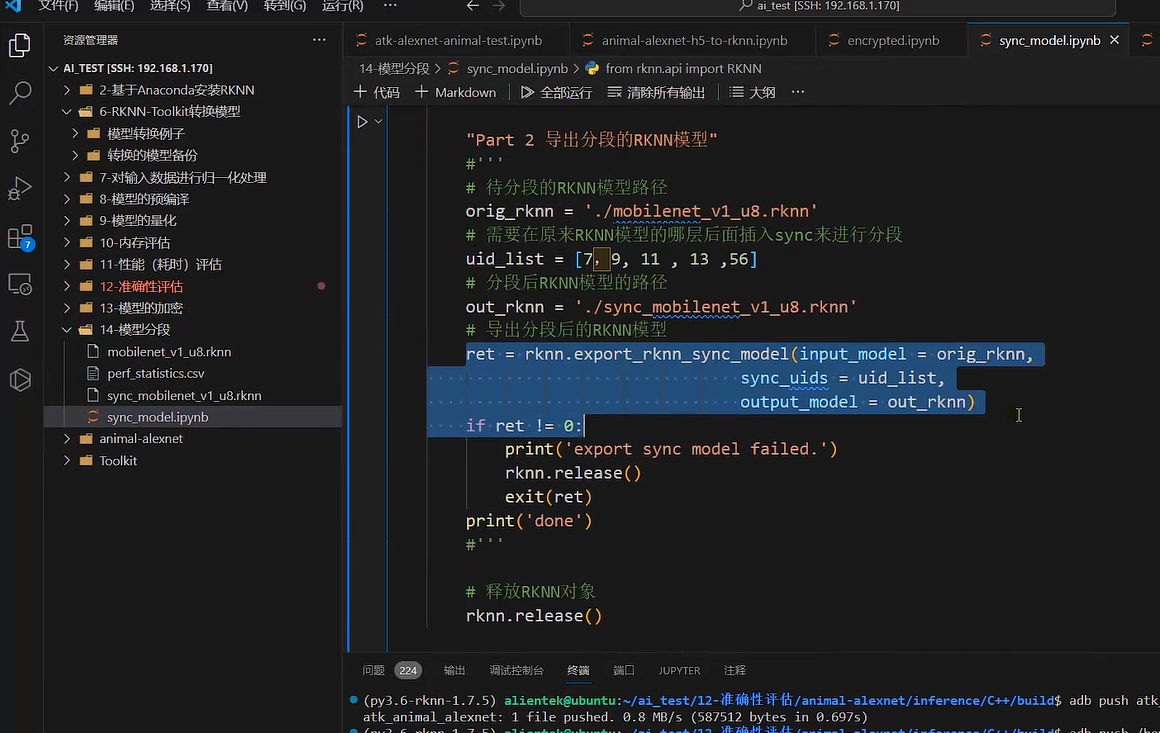
如果不对模型进行分段，假设在初始化的时候，先初始化模型A，再初始化模型B，那么模型推理时的顺序就是上图那样。先推理模型A再推理模型B。

接下来假设要对模型A进行分段，分别分为A1和A2两端。模型B不进行分段。那么推理的时候就如上图：先推理模型A的前半段，再推理模型B，推理了模型B之后，再推理模型A的后半段。也就是A2。

\*/

/\*

下面看案例代码，实现对模型的分段操作：



导出分段的模型的api就是图中那个。它有3个参数：

Input\_model：就是待分段的RKNN模型的路径。

Sync\_uids:就表示要在原来的RKNN模型的哪个层后面进行分段，这个参数是一个列表的形式，列表里写的就是 模型某个层的uid号。

Output\_model：就是导出的分段的RKNN模型的路径。

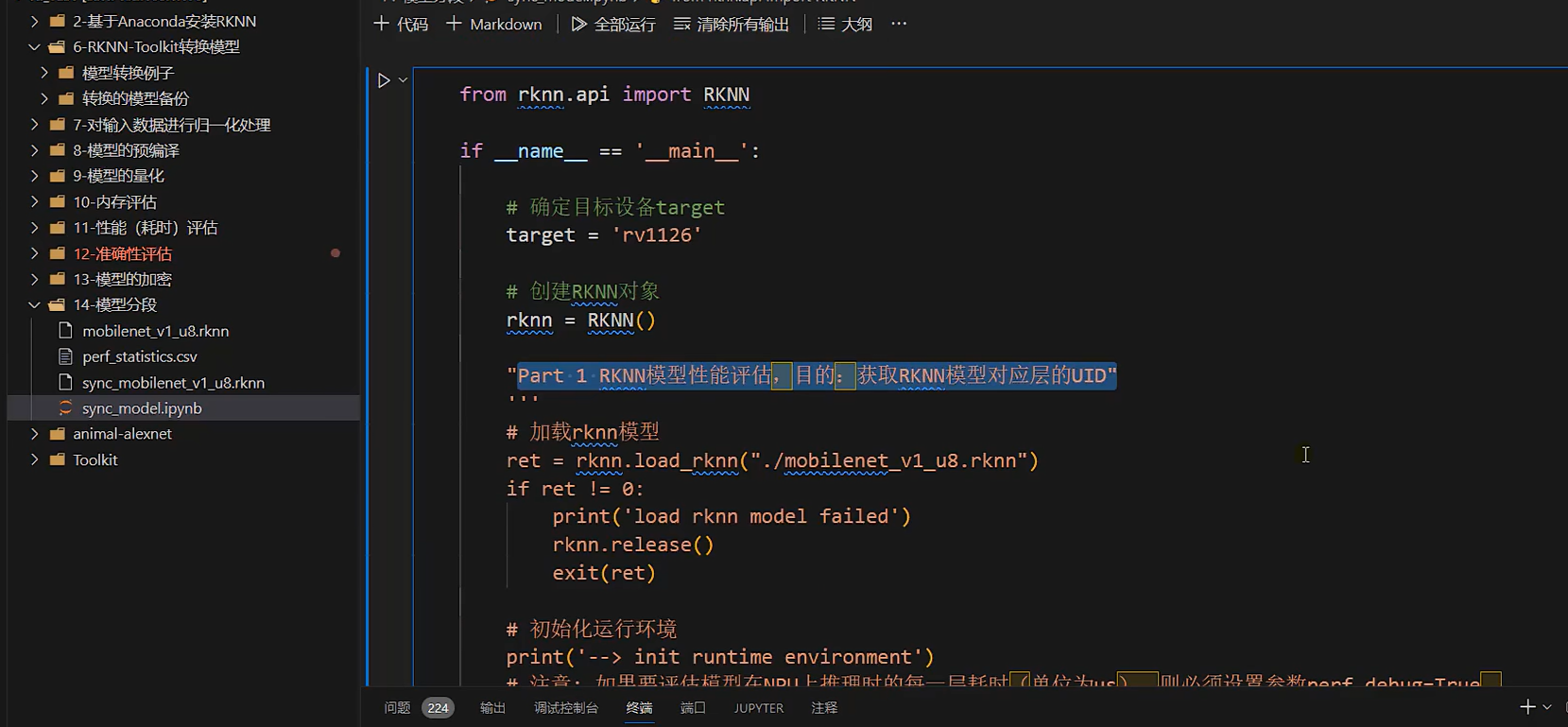
如果要指定参数里的uid，这个uid的获取方式就是：通过模型性能评估的接口来获取 层的uid号。

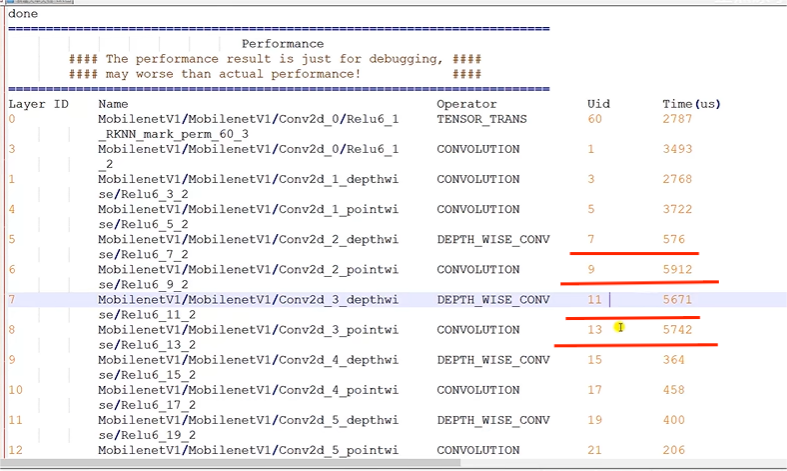
\*/

/\*

代码的运行方式：

首先运行第一部分的代码，也就是对模型进行性能评估获取uid号，此时需要将第二部分的代码注释掉。先运行前面的代码。进行性能评估的时候，注意要ADB连接开发板。

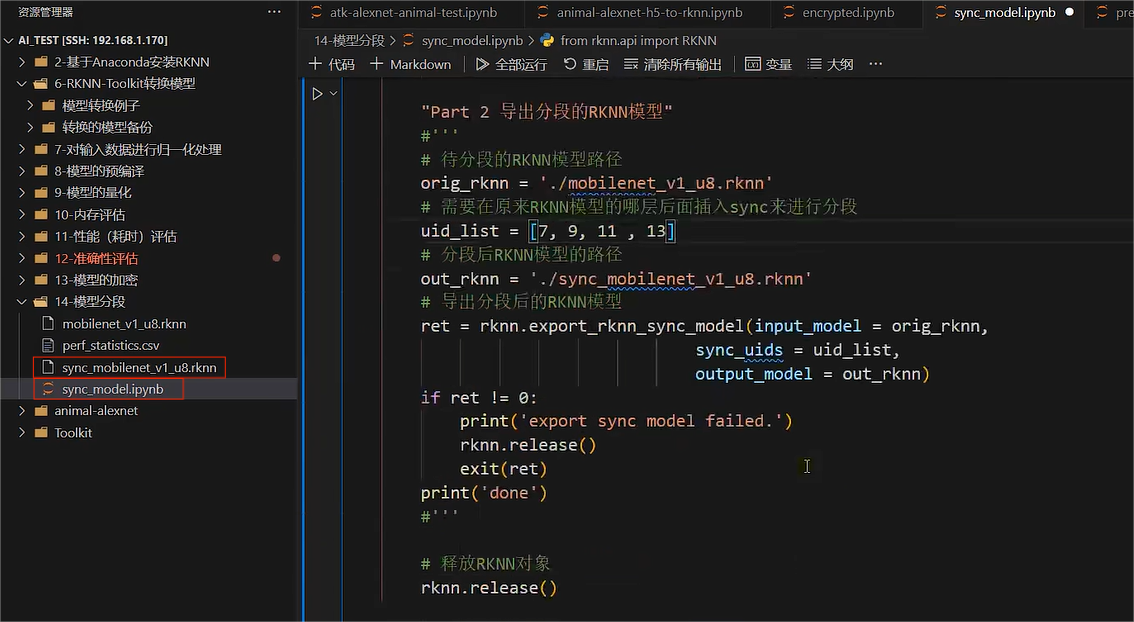




通过运行的结果可以看到哪些层的耗时比较长，我们就在这些层进行分段，这样模型就被分成了5段了。

确定好uid号之后，我们就运行第二部分的代码，把第一部分的代码注释掉（第一部分的代码主要就是完成获取uid号的），第二部分就是导出分段的模型：

我们是将这个模型从uid的后面分成了5段：



导出的分段模型就是上图的：v1\_u8.rknn模型。

RKNN Toolkit封装好的这些api给我们调用，我们直接使用即可，实现起来还是比较简单的。

\*/

\*/