

本周先把 `proposal_target_layer` 层写完，在 `roi pooling` 层时 `faster rcnn` 的代码使用的 `torch.utils.ffi` 库在 `pytorch1.0` 版本被删除了，官方推荐使用 `cpp_extension` 库来进行 C++ 的扩展。用 `cpp_extension` 没有成功完成 `build`，可能是连接函数没有写好，这个除了官方给出的 `doc` 以外在 `github` 找到了一份示例 <https://github.com/chrischoy/pytorch-custom-cuda-tutorial>，准备后续搞明白后再试试，后面尝试用 `cffi` 库的扩展，但因为 C 编译器本身不支持 `tensor` 所以无法成功。我自己尝试在 `torch.autograd.Function` 的继承类中通过 `adaptive max pool` 写了 `roi pool` 的相关算法，在 `backward` 中使用的是把 `grad_output` 对应每个 `roi` 的相关索引的值都放到一个 `grad_input` 对应的索引位置，不确定是否正确。同时也在 `torch.nn.Module` 的继承类中通过 `for` 循环写了个 `roi_pooling` 层。

之后把训练的相关代码完成了，目前正在进行调试工作，`rpn` 层的有部分的 `numpy` 和 `tensor` 的相关操作混乱导致了很多错误，正在把能把 `numpy` 改成 `tensor` 的地方都做修改，计划这部分结束之后测试一下自己的两种 `roi pooling` 层的代码是否有错，然后去搞 `CUDA` 部分的 `roipooling` 代码。