

1. 중간에 ReLU나 Dropout은 생략되어도 정답으로 인정할 수 있을 것 같습니다. Conv2d는 *input_channels, output_channels, kernel_size, stride, padding*으로 512->4096->4096 이 정답이긴 하나, 512->2048->2048 같은 임의의 수로 맞추어도 문제는 없을 것 같습니다.

```
self.fconn = nn.Sequential(  
    nn.Conv2d(512, 4096, 7, 1, 3),  
    nn.ReLU(inplace=True),  
    nn.Dropout(),  
    nn.Conv2d(4096, 4096, 1),  
    nn.ReLU(inplace=True),  
    nn.Dropout(),  
)
```

2. pseudo code로 작성되어 의미만 비슷하면 답으로 인정할 수 있을 것 같습니다.
 1. tracking할 bbox주변에서 samples들을 추출한 이 sample들로 다음 bbox regressor를 업데이트 합니다. 그 후 bbox regressor를 사용해 predict한 bbox를 다음 bbox로 생각합니다.
 2. tracking할 bbox를 찾지 못했기 때문에 이전에 있던 bbox를 그대로 다음 bbox로 생각합니다.
 3. Short Term Update: 트래킹에 실패할 경우 짧은 시간 내로 모아진 pos, neg features로 네트워크를 훈련합니다
 4. Long Term Update: 트래킹에 성공할 경우는 임의의 시간간격 (문제에서는 `opts.long_interval`) 으로 pos, neg features로 네트워크를 훈련합니다.