1.

VGG16 accuracy:

```
criterion = nn.CrossEntropyLoss().cuda()

model.eval()
model.cuda()


prec = validate(testloader, model, criterion)
```

```
Test: [0/79]    Time 0.154 (0.154)      Loss 0.2583 (0.2583)    Prec 92.969%
(92.969%)
 * Prec 90.280%
```

Resnet20 accuracy:

```
model.cuda()


prec = validate(testloader, model, criterion)
```

```
Test: [0/79]    Time 0.160 (0.160)      Loss 0.2343 (0.2343)    Prec 94.531%
(94.531%)
 * Prec 91.150%
```

2.

Prehook (VGG16):

```
]: torch.Size([128, 3, 32, 32])
```

```
]: con = model.features[0]
   Norm = model.features[1]
   Rel = model.features[2]
```

```
]: my_output = Rel(Norm((con(my_input))))
```

```
]: (my_output - save_output.outputs[1][0]).sum()
```

```
]: tensor(0., device='cuda:0', grad_fn=<SumBackward0>)
```

Prehook (Resnet20):

[34]:
```
res = my_input
my_output = Rel((bn_2(conv_2(Rel(bn_1(conv_1(my_input))))))+res)
```

[35]:
```
(my_output - save_output.outputs[3][0]).sum()
```

[35]:
```
tensor(0., device='cuda:0', grad_fn=<SumBackward0>)
```