

1.pytorch version:1.0.0

2.The training argument of the classifier:

Batch size: 4

Learning rate:0.001

Optimization method: SGD(momentum=0.9)

Epoch number:6

3.Accuracy of the network on the 10000 test images: 61 %

Accuracy of plane : 54 %

Accuracy of car : 91 %

Accuracy of bird : 31 %

Accuracy of cat : 41 %

Accuracy of deer : 50 %

Accuracy of dog : 54 %

Accuracy of frog : 72 %

Accuracy of horse : 76 %

Accuracy of ship : 72 %

Accuracy of truck : 72 %

4.The tracking testing result(AUC):With loading classifier parameters, accumulating w and x, the result AUC is 0.5100.

With standard tracking model, the result is 0.6388

5.When we run `loss.backward()`, pytorch will compute gradients of every notes and add these gradients to their notes. And when we run `optimizer.step()`, the optimizer we define(SGD) will upgrade all the weights in these notes according to the gradients computed before.

6.First, we get a filter W according to input X and idea gaussian Y . The transformation of X correlated with this filter W (which is called response) will get a high response in the center of our target object. Since we assume this target object will move slowly in a video, the X will changes slowly. When there comes a new frame, we use the filter we get from last frame to correlate with patches from current frames, and we think the location of max of these new responses will be the location of our target object in the new frame. Again, we use the new location of target object and idea gaussian to compute a new W and update the old W with the new W .