

# Assignment1 of Artificial Intelligence

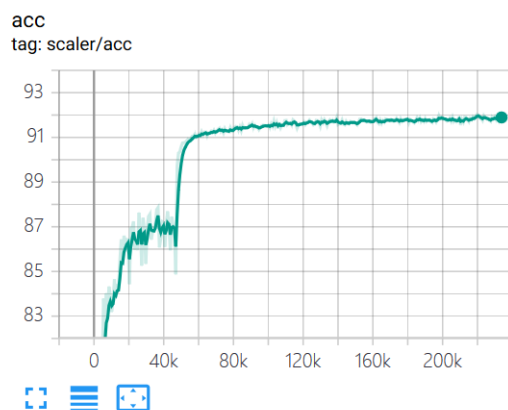
马奕/1901213135

## Create model and train

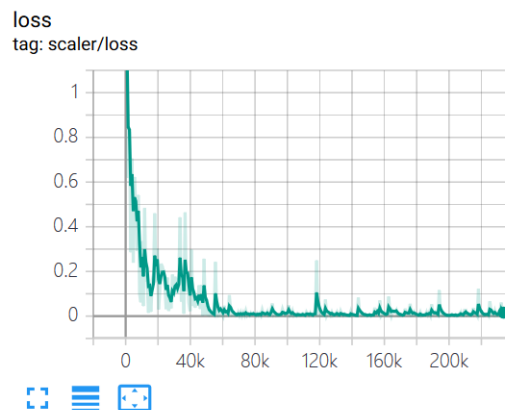
First, I built an 18-layer network with reference to ResNet18, and then trained it according to the following parameters:

- BATCH SIZE = 64
- EPOCH = 300
- LR = 0.01 (or 0.001 for epoch>60)
- Loss : CrossEntropyLoss()
- Optimizer : SGD

## Training result:



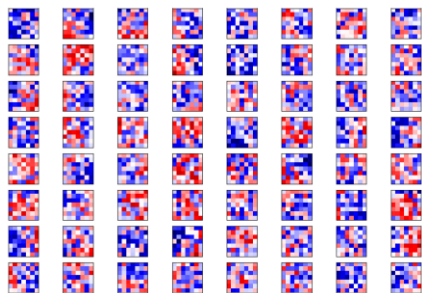
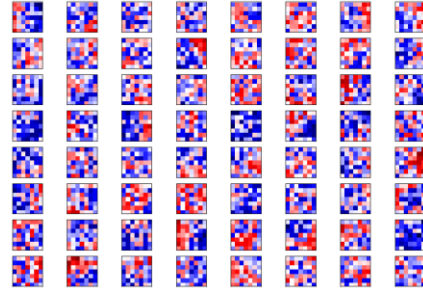
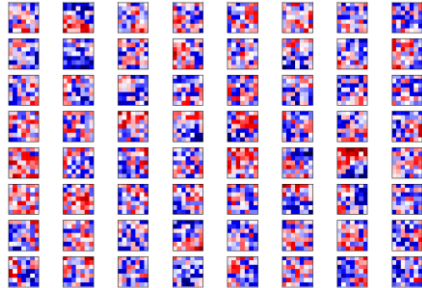
Accuracy of Test Set : 91.89%



Loss of Test Set : 0.0066

## Filter visualization

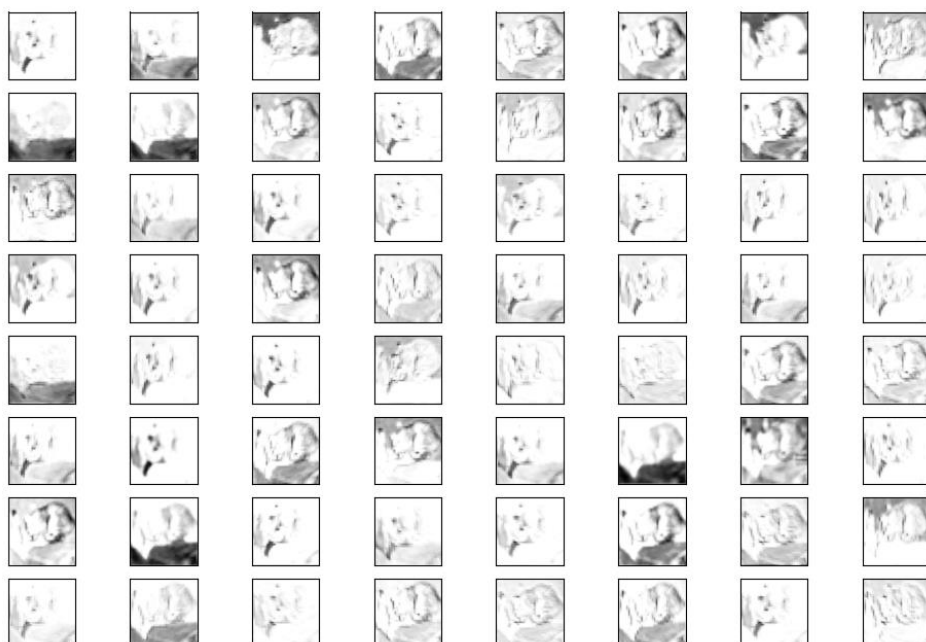
Weight of conv1 ,kernel size=(7\*7), stride=2 , channel=1~3:



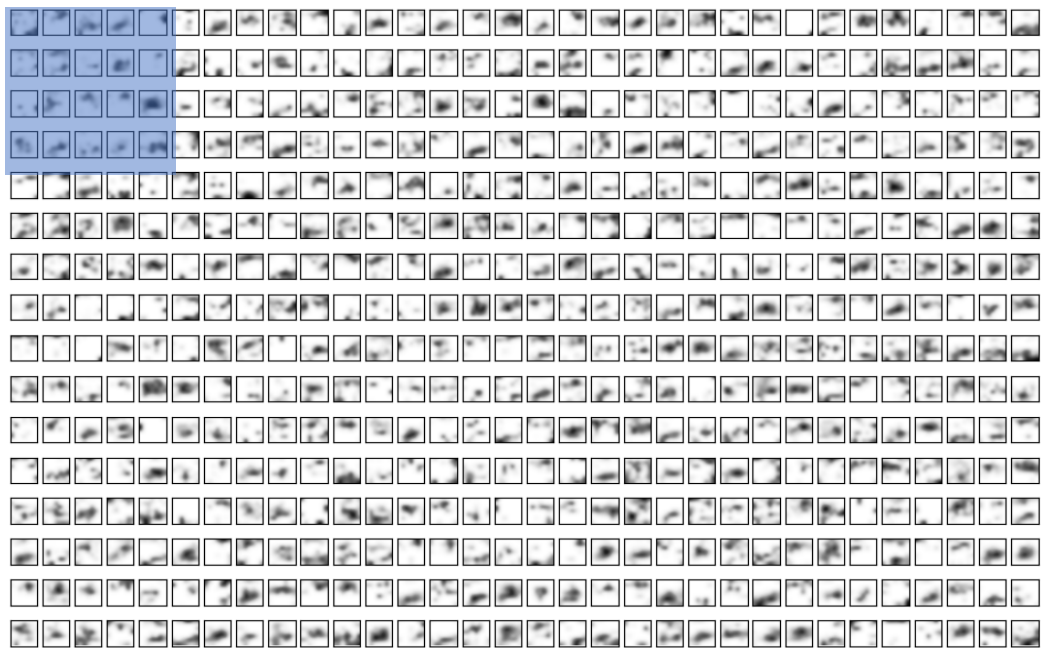
## Feature maps visualization

The original image is in the next section

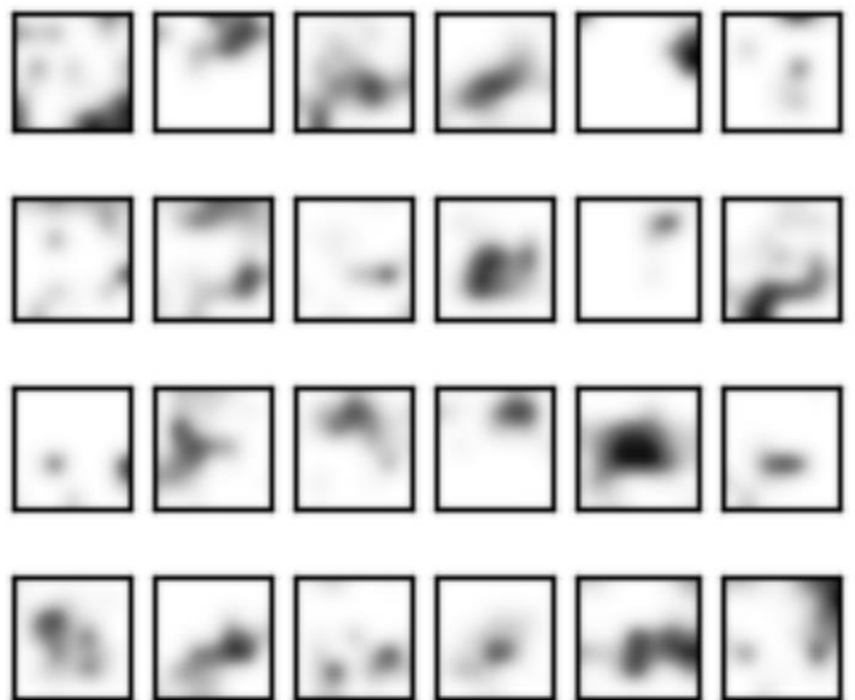
Feature maps for Conv1:



Feature maps for Conv5\_x:



Some details of the figure above:

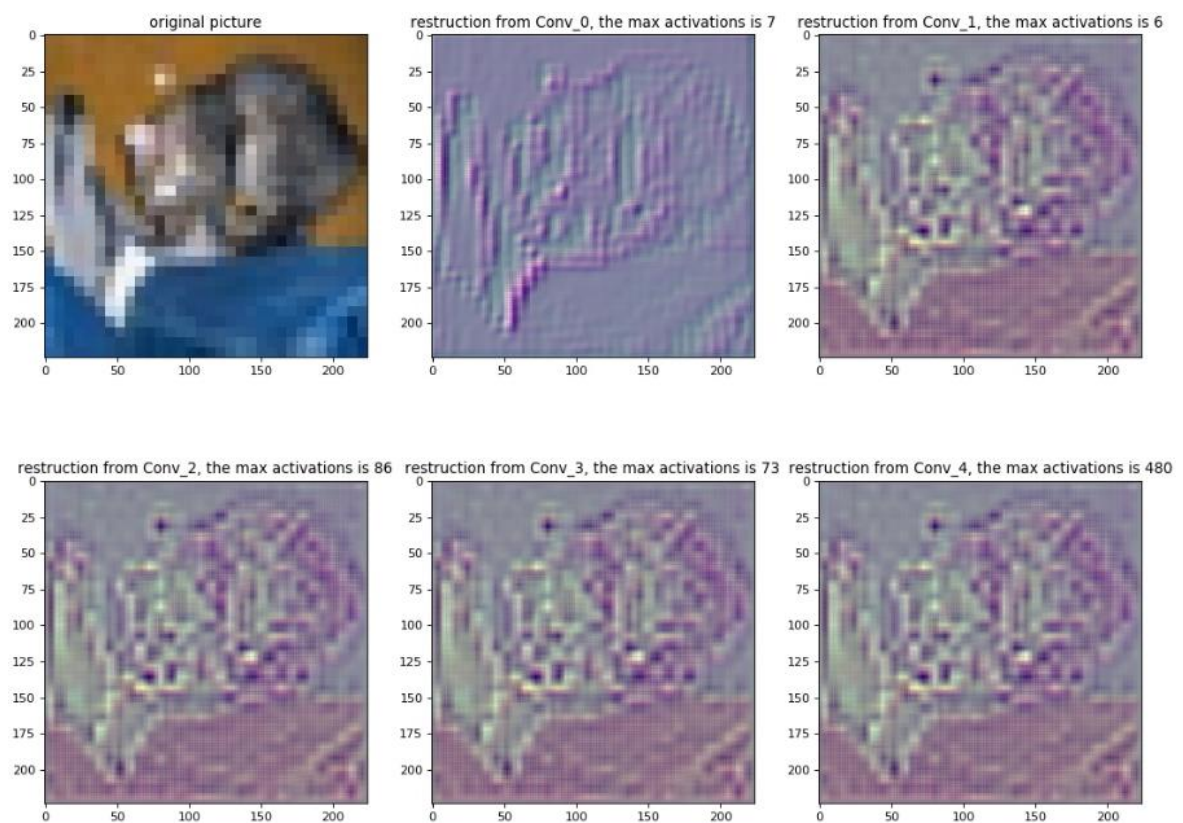


## Reconstruction

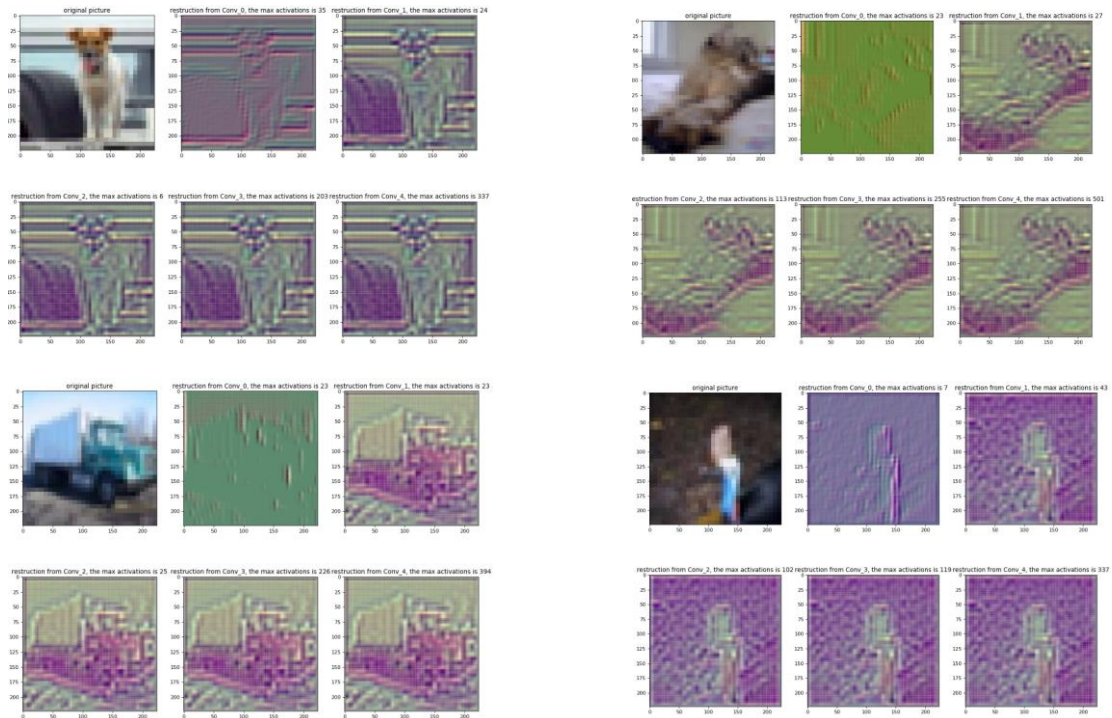
The reconstruction of the feature map that has the maximum activation value.

In this task, I first look for feature map with the maximum activation value for each layer, then pass it through the reverse network of ResNet18.

To build a reverse network, I use the `ConvTranspose2d ()` to reverse `Conv2d ()`, for Residual Block in the ResNet18 and its reverse, the shortcut must be matched. I do reconstruct for the feature map of every layer, this is the result :



And for other images:



The reconstruction of layer3~5 is not much different , I think it is caused by the shortcut in the network.

## Feedback

- The assignment took me about a week.
- The assignment is challenging, I enjoy it.
- At the beginning of the course, I thought small assignments would be better.

## Reference

res18\_train.py

- [1] <https://blog.csdn.net/sunqiande88/article/details/80100891>

deRes.py, res18\_view.py

- [1] <https://github.com/huybery/VisualizingCNN>

[2] <https://github.com/grishasergei/conviz>