Assignment 1 CNN for image recognition

Ke Qiu

1901213138, qiuke@pku.edu.cn,

Report the first assignment of the AI course, 2019.10

1. Experiments Details

1.1 Introduction

In this project, I build a standard Resnet18 network based on Fig.2 in the specified reference paper[1]. At the same time, I fully realized the requirements of the project, including network construction, optimization and visualization. This report will introduce my experimental method, experimental results and specific analysis in detail. Based on Windows and PyCharm, I use Cifar10 dataset to train my network . Thanks to a powerful gtx1060ti, I have tried many aspects of the project quickly, and familiar with a lot of knowledge and skills about AI.

1.2 Dataset processing

- Divide the data. I used 50000 pictures from cifar10 for training and the remaining 10000 for testing.
- Data processing. Before the data entered the model, I tried to perform a variety of combination operations on the training data, including mix train data, random clipping, random horizontal flipping, image Normalization, and resize image to (224, 224, 3). As shown in Table 2, the experimental results show that the random operation can significantly improve the over fitting. In addition, enlarging images to 224 × 224 helps improve test accuracy. This scheme can make the accuracy more than 90% in my model. But I don't think it makes sense. It's better to use Cifar100 than enlarge Cifar10.

1.3 Resnet18

As shown in Table 1, my network structure is very challenging. The network input is very small, only 32×32 big. This standard network will make Con5-x output 512 points.

Layer Name	Output Size	Paremetes : kernel, Stride, Channel
Conv1	**16×16**	**7×7, stride 2, 64**
Con2-x	**8×8**	**3×3** max pool, 64 , 2 $\begin{bmatrix} 3\times3, 64 \\ 3\times3, 64 \end{bmatrix}\times2$
Con3-x	**8×8**	**3×3** , 128, 1 $\begin{bmatrix} 3\times3, 128 \\ 3\times3, 128 \end{bmatrix}\times2$
Con4-x	**4×4**	**3×3** , 256, 2 $\begin{bmatrix} 3\times3, 256 \\ 3\times3, 256 \end{bmatrix}\times2$
Con5-x	**2×2**	**3×3** , 512, 2 $\begin{bmatrix} 3\times3, 512 \\ 3\times3, 512 \end{bmatrix}\times2$
	1×1	average pool , 10-d FC, softmax

Table 1. Structure of My standard Resnet18

1.4 Train Parameters

- **Parameters:** Batchsize = 100 ; epoch = 20; step = 1000;
- **Optimizer:** Adam, Initial Learning Rate = 0.001, milestones=[10, 15, 18], gama = 0.4
- **Loss:** CrossEntropyLoss

In order to optimize the network, I have try my best to adjust the parameters.

Here are some examples in Table2:

Batch size	milestones	gama	Preprocessing	Test_ACC(%)	Train_ACC(%)
100	[10, 15]	0.122	√	83.98	87.43
100	[10, 15, 18]	0.5	√	84.18	88.39
100	**[10, 15, 18]**	**0.4**	**√**	**84.82**	**88.33**
100	[10, 15, 18]	0.4	×	82.30	99.45
100	[10, 15, 18]	0.3	√	84.35	88.20
100	[8, 10, 15]	0.4	√	81.62	85.64
100	[8, 10, 15,18]	0.5	√	83.92	87.24
100	[8, 10, 15,18]	0.4	√	83.67	87.29
200	[10, 15, 18]	0.4	√	83.58	88.15
			...		

Table 2. some examples to adjust parameters

2. Research Results

2.1 Train Results

- **Final train accuracy:** 88.3% (step10000)
- **Final test accuracy:** 84.8% (step10000)
- **Final train accuracy:** 90.76% (step20000)
- **Final test accuracy:** 85.1% (step20000)

2.2 Learning progress visualization

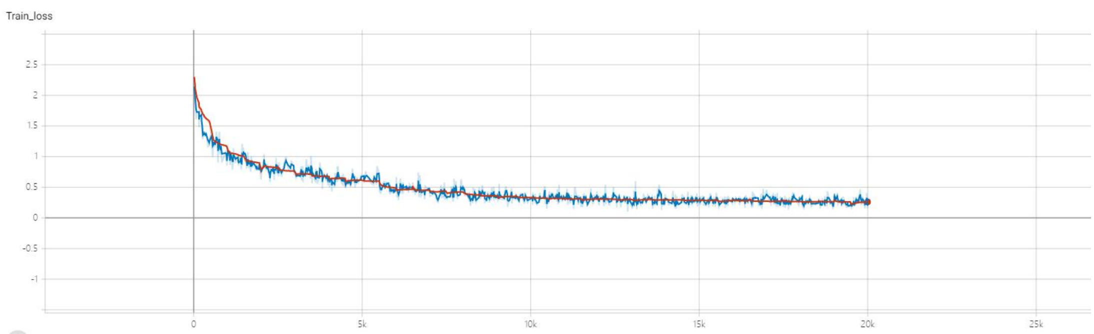


Fig 1. Train Loss(cross_entroy) (from step =0>>20000)

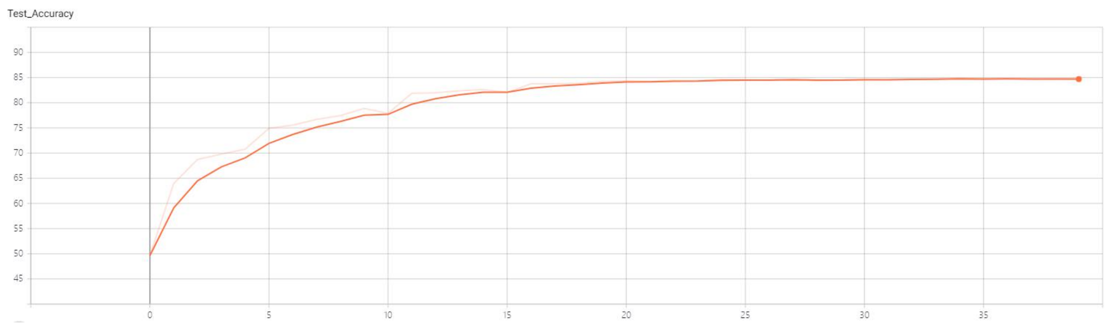


Fig 2. Test_Accuracy (from epoch=0>>20)

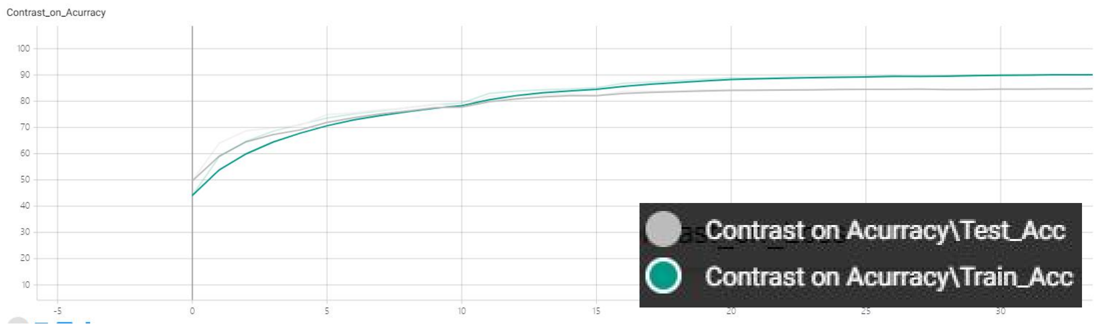


Fig 3. Train/Test_Accuracy Contrast (from epoch=0>>20)

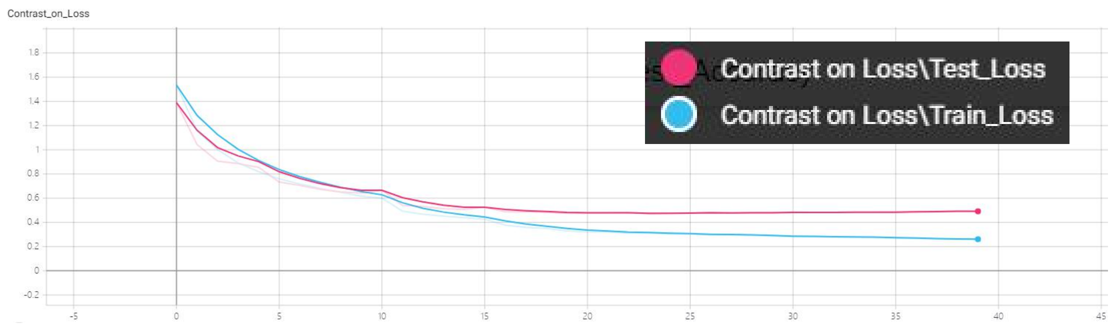
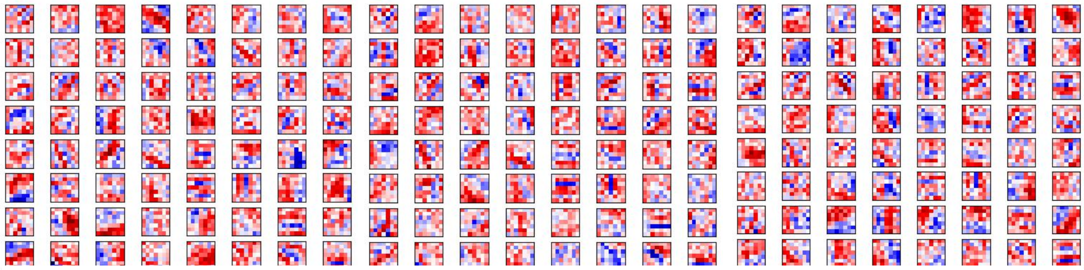


Fig 4. Train/Test_Loss Contrast (from epoch=0>>20)

As shown in **Figure 1**, my network is close to convergence in the 10th epoch, and it doesn't converge in the 15th epoch. This proves that my network converges well with only 20 iterations. Next, it's time to modify the network's structure and loss instead of parameters. Similarly, **Figure 2** also proves that my network **quickly** converges well with **only 20 epoch**.

In addition, **Figure 3&4** show that My network has **no serious over fitting** phenomenon. Before the network close to convergence, there is no significant difference between training and testing in terms of loss and accuracy.

2.3 Filter visualization



(a) Channel-1 (b)channel-2 (c)channel-3

Fig 5. Filter visualization of Conv1 (step = 10000)

Method 1: The layer 'Conv1' contains 64 different filters, which have 3 channels. The size of each filter is 16×16×3. In order to visually compare these filters, I divided them into three channels to show. Refer to conviz’s method[2], I showed them in specific color patterns. Obviously, Red and blue represent the two extremes of filters’ weights.

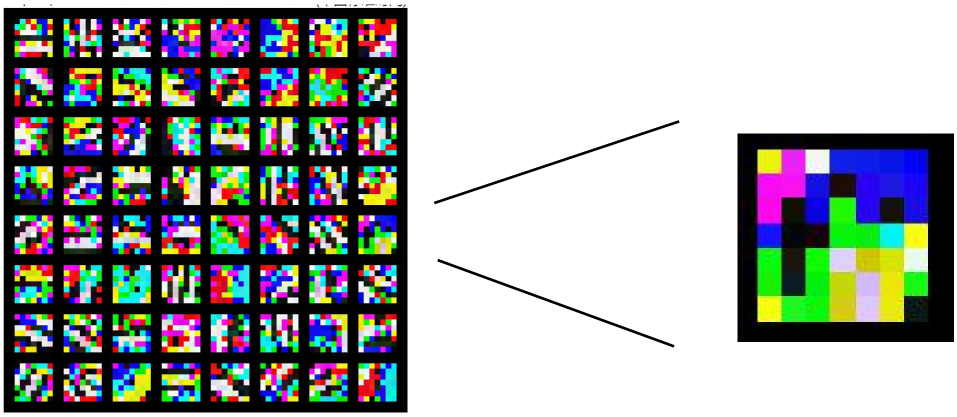


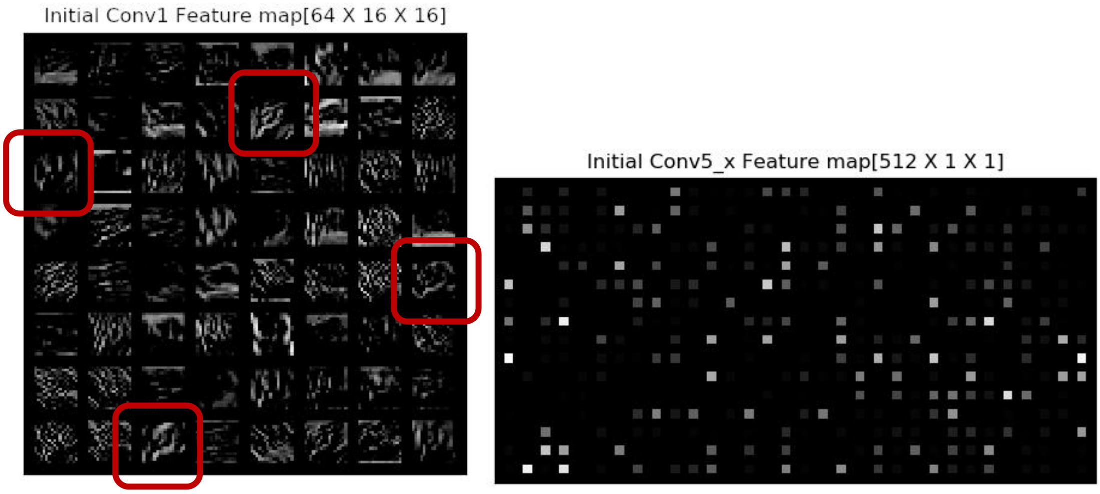
Fig 6. Filter visualization of Conv1

Method 2: Refer to conviz’s method[2], I showed filter in specific color patterns. Based on tensorboard and torchvision, I fuse the three channels of weight. So, I can visualized the Conv1’s filter In the form of RGB.

2.4 Feature mapping visualization

2.4.1 Initial Feature map from Conv1, Conv5_x

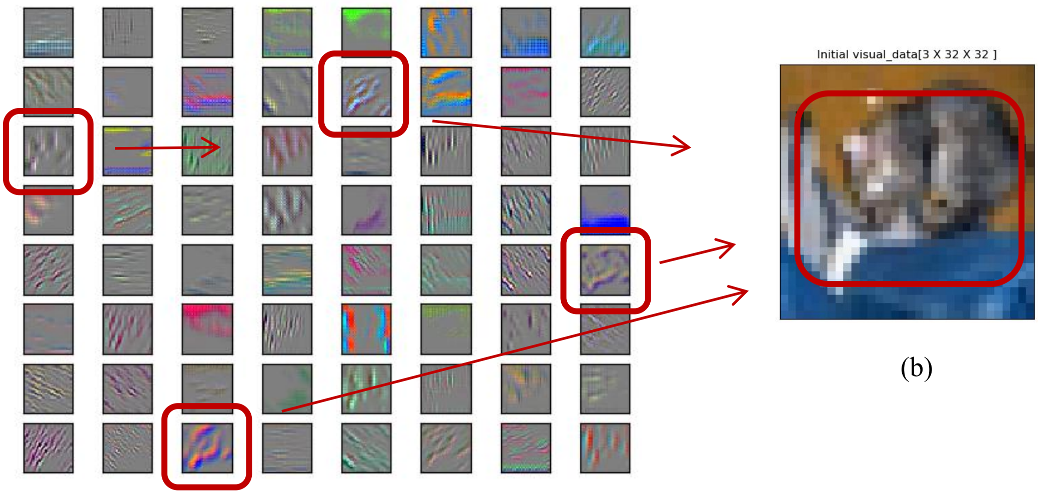
To compare the reconstructed feature-maps, Let's first show the feature map extracted directly from conv1 and Conv5. As you can see, they are grayscale images. Their sizes are [64×16×16] and [512×1×1].



(a)Initial Conv1 Featuremap (b)Initial Conv5_x Featuremap

Fig 7. Feature map visualization of Conv1(a) and Conv5_x(b)

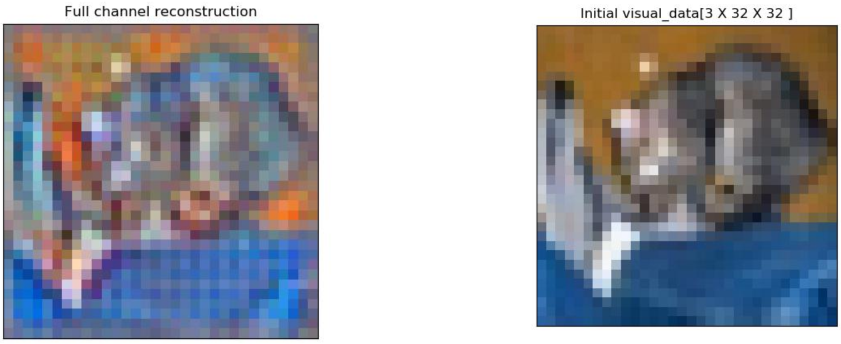
2.4.2 Reconstructed Feature map from Conv1, Conv5_x



(b)

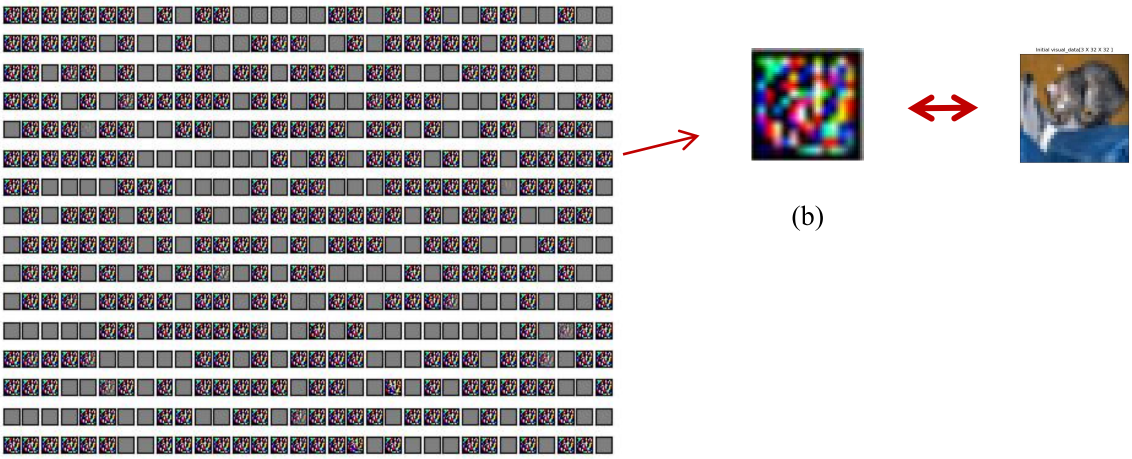
(a)

Fig 8. 64 Reconstructed Feature map from Conv1(a) , input image(b)



(a) (b)

Fig 9. Reconstructed entire Feature map from Conv1(a) , input image(b)



(b)

Fig 10. Reconstructed entire Feature map from Conv5_x(a) , one map(b)

As shown in Figure 7, I extracted the output of conv1 and conv5 X in resnet18 directly. The visual effect of the feature map(a) extracted directly is not good. Intuitively, it can only reflect the extracted texture information. The visual effect of the feature map(b) is very bad, it can reflect nothing.

As shown in Figure 8, In order to reconstruct images, I write a **reverse Resnet18** , which is used to calculate the feature map. This **reverse Resnet18** is the opposite of the previous Resnet18, which is composed of transposition convolution, Maxunpooling, shortcut, etc. For the output of Conv1 and Conv5_x, I only take the data of **one channel at a time** and set the others to zero. So, Conv1 reconstructed 64 images and Conv5_x reconstructed 512 images .

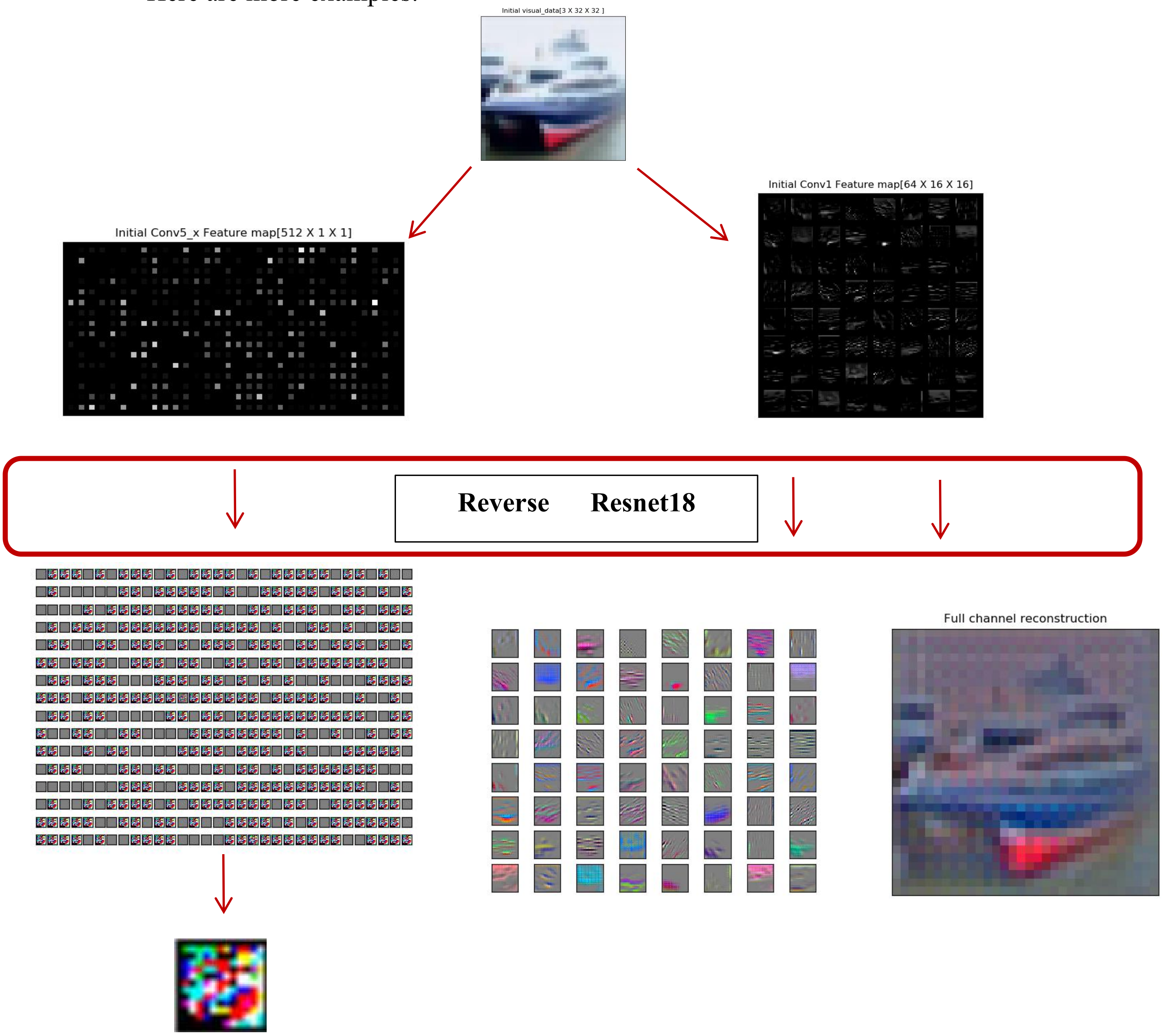
Based on this method, given a feature map[B, C, H, W], I can reconstruct C Feature images. Then, i chose some reconstructed patterns from the validation set that cause

high activations in a given feature map. I've **circled it in red** in Figure 8.

Interestingly, when i found some reconstructed patterns that cause high activations, it’s easy to find the Corresponding patterns in original feature map. As you can see in red box, this layer (Conv1) mainly extracts the texture information of the image.

As shown in Figure 9, I directly import the complete feature map into **reverse Resnet18** for reverse reconstruction. Obviously, it lost some information, including texture and color. As shown in Figure 10, It's too hard to find the corresponding pattern from the reconstructed image. Sometimes, i can get some abstract information. And i think, the image reconstructed from Conv5_x loses too much information and has no reference value.

Here are more examples:



3. Discussion

- Cifar10 is small(32×32). The output size of Conv5_x is 1×1. On the one hand, it is not suitable for so many convolutions. On the other hand, it also affects the effect of visualization.The next step maybe use Cifar100 or big size dataset to promote the effect of visualization and the value of accuracy.
- When build the Reverse Resnet18, it hard to solve the problem of shortcut. Intuitively, it's not reasonable to add or subtract shortcut directly. It was worth to further study

4. Feedback

- **Time your spend for this assignment, i.e., how many hours?**

  Nearly a week.Working hours are more than 40 hours.

- **Comments for this course?**

AI is a very important course for me. Many of my courses are related to it. In the course, I have learned some basic knowledge as well as some new research ideas. I hope I can learn more experience, ideas and skills.

- **Comments for this assignment?**

This work is very challenging, especially the visualization part. For me, I feel a little nervous about the assignment deadline. In fact, if I was given more time, I believe I can learn more and try more.

- **Suggestion for the following lectures?**

As everyone knows, we have some similar course like CV, Machine Learning,etc. In my opinion, AI is the broadest and highest-level kind of course. I hope this course is different from these similar courses. Recently, I have been reading some books named *artificial intelligence.* In the following lectures, I hope to learn AI systematically.

5. References

1. Paper :Deep Residual Learning for Image Recognition:

   Link: https://arxiv.org/pdf/1512.03385.pdf

2. Code: Conviz.py, Utils.py

   a) Reference link: https://github.com/grishasergei/conviz
   b) The function: Plot_visual() refer to both of the plot_conv_weights