Report of Experiment 5&6

1. Intro to ResNet50

Residual net is a kind of Deep convolutional neural networks with a depth of up to 152 layers---8x deeper than VGG nets. It involved the Residual "Representations" and "Shortcut Connections" which allowed the training of the deep net become possible.
The structure of ResNet50 is shown as blow:

layer name	output size	18-layer	34-layer	50-layer	101-layer	152-layer
conv1	112×112	7×7, 64, stride 2				
conv2_x	56×56	3×3 max pool, stride 2				
		$\begin{bmatrix} 3\times3, 64 \\ 3\times3, 64 \end{bmatrix}\times2$	$\begin{bmatrix} 3\times3, 64 \\ 3\times3, 64 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 64 \\ 3\times3, 64 \\ 1\times1, 256 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 64 \\ 3\times3, 64 \\ 1\times1, 256 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 64 \\ 3\times3, 64 \\ 1\times1, 256 \end{bmatrix}\times3$
conv3_x	28×28	$\begin{bmatrix} 3\times3, 128 \\ 3\times3, 128 \end{bmatrix}\times2$	$\begin{bmatrix} 3\times3, 128 \\ 3\times3, 128 \end{bmatrix}\times4$	$\begin{bmatrix} 1\times1, 128 \\ 3\times3, 128 \\ 1\times1, 512 \end{bmatrix}\times4$	$\begin{bmatrix} 1\times1, 128 \\ 3\times3, 128 \\ 1\times1, 512 \end{bmatrix}\times4$	$\begin{bmatrix} 1\times1, 128 \\ 3\times3, 128 \\ 1\times1, 512 \end{bmatrix}\times8$
conv4_x	14×14	$\begin{bmatrix} 3\times3, 256 \\ 3\times3, 256 \end{bmatrix}\times2$	$\begin{bmatrix} 3\times3, 256 \\ 3\times3, 256 \end{bmatrix}\times6$	$\begin{bmatrix} 1\times1, 256 \\ 3\times3, 256 \\ 1\times1, 1024 \end{bmatrix}\times6$	$\begin{bmatrix} 1\times1, 256 \\ 3\times3, 256 \\ 1\times1, 1024 \end{bmatrix}\times23$	$\begin{bmatrix} 1\times1, 256 \\ 3\times3, 256 \\ 1\times1, 1024 \end{bmatrix}\times36$
conv5_x	7×7	$\begin{bmatrix} 3\times3, 512 \\ 3\times3, 512 \end{bmatrix}\times2$	$\begin{bmatrix} 3\times3, 512 \\ 3\times3, 512 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 512 \\ 3\times3, 512 \\ 1\times1, 2048 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 512 \\ 3\times3, 512 \\ 1\times1, 2048 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 512 \\ 3\times3, 512 \\ 1\times1, 2048 \end{bmatrix}\times3$
	1×1	average pool, 1000-d fc, softmax				
FLOPs		1.8×10^9	3.6×10^9	3.8×10^9	7.6×10^9	11.3×10^9

Table 1. Architectures for ImageNet. Building blocks are shown in brackets (see also Fig. 5), with the numbers of blocks stacked. Down-sampling is performed by conv3_1, conv4_1, and conv5_1 with a stride of 2.

(picture from [1])

2. Experiment 5 (Transfer Learning)

First the ResNet50 model has been loaded form Kears.Application, the commends as blow:

```
model = keras.applications.ResNet50(include_top=False, weights='imagenet', input_tensor=None, input_shape=(32, 32, 3), pooling=None, classes=1000)
```

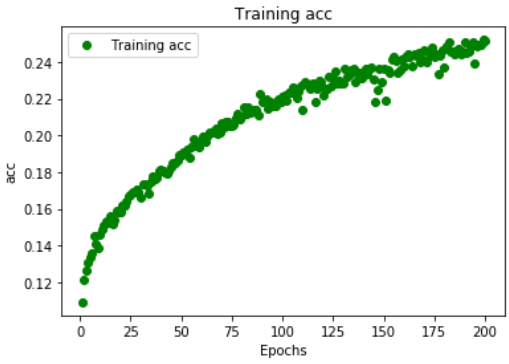
The input shape has been modified to 32*32 in order to fit CIFAR10's image size. The top 3 fully connected layers of the model has been discard to leave places for the following 10-Class classifier. However, the weight of the deep Convolution-Layers are keep.

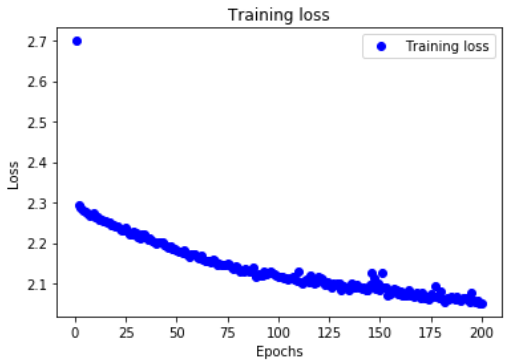
The parameters of the model:
Total params: 23,587,712
Trainable params: 23,534,592
Non-trainable params: 53,120

The ImageDataGenerator method with batch size 4 has been applied to augment the original CIFAR-10 training set. The original training set has 50000 examples and the test set size is 10000.
Both the training set data and test data are passed through the ResNet50's model and the bottleneck features are obtained and restored to files for later classification. The epoch was set to 50

Finally, the training accuracy and losses are obtained as follows:

test_acc: 0.1
train acc: 0.09987834662724997





3. Experiment 6 (Fine-tuning)

The ResNet50 model is loaded from Keras as in expriment5. Then the last 7 layers are remained trainable while the previous 170 layers' weight are been switched to fix.

The parameters of the fine-tuning model:
Total params: 23,587,712
Trainable params: 3,415,552
Non-trainable params: 20,172,160

Then, a sequential dense model are added to train the model. The structure of the full model:
top_model.summary()

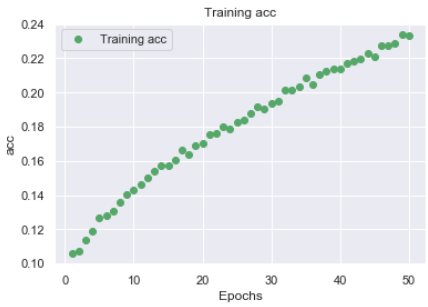
Layer (type)	Output Shape	Param #
resnet50 (Model)	(None, 1, 1, 2048)	23587712
flatten_6 (Flatten)	(None, 2048)	0
dense_10 (Dense)	(None, 512)	1049088
dropout_7 (Dropout)	(None, 512)	0
dense_11 (Dense)	(None, 256)	131328
dropout_8 (Dropout)	(None, 256)	0
dense_12 (Dense)	(None, 10)	2570

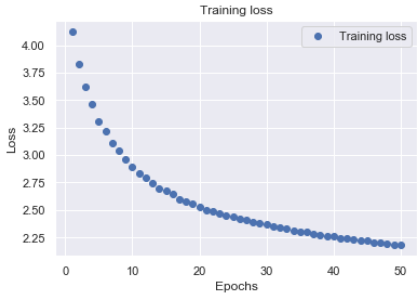
Total params: 24,770,698
Trainable params: 4,598,538
Non-trainable params: 20,172,160

The whole training process is stick to the principle that low learning rate and SGD optimizer.
The epoch was set to 50

```
top_model.compile(optimizer=optimizers.SGD(lr=1e-4, momentum=0.9),
          loss='categorical_crossentropy',
          metrics=['accuracy'])
```

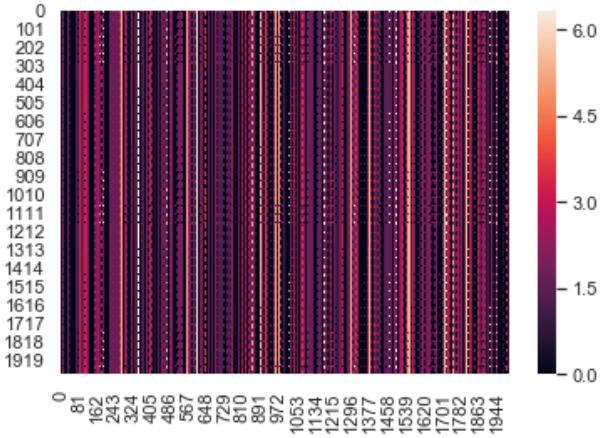
Results:
test_acc: 0.1009
train acc: 0.23333999994754792



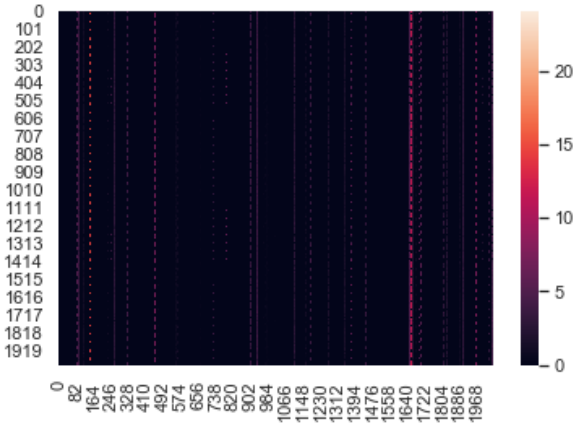


4. Disscusion

Far from expected, the accuracies of the two experiment are pretty low. The potential reason could be the Convolution layers. As the weights of the model are trained by 224*224 image data, it is reasonable that it could be failed on 32*32 dataset. First, the convolutional kernel of the first layer is 7*7 which could be too large for images with less pixels. Besides, the weights optimized by high resolution images may not extract useful features when applied on low resolution data.
In order to see the efficacy of the Convolution net, we visualized the outputs:
The output of layer 7 (After first convolute operation, randomly sampled 2000):



The bottleneck features (randomly sampled 2000):



As the pictures showed, the discrepancies among samples are vanishing after first convolute operation.

It is not to say that the low-resolution data like 32*32 with 10 class is not fit for deep net, As in work of Kaiming He, Xiangyu Zhang etc.[1] , the Deep Residual net could also get accuracy over 90%. The flowing hyper-parameters in the article lead to a pretty high accuracy:
The first layer is 3 × 3 convolutions. Then a stack of 6n layers with 3 × 3 convolutions on the feature maps of sizes { 32, 16, 8 } respectively, with 2n layers for each feature map size. The subsampling a stride of 2.

According to all above, a possible way to higher the power of the model is retrained all the weight on 32*32 data base with modified model structure.

5. Reference

[1] Kaiming He, Xiangyu Zhang, Shaoqing Ren, Jian Sun Deep Residual Learning for Image Recognition