Selected Topics in Visual Recognition using Deep Learning Homework 1

Jia-Wei Liao, 309652008
Department of Applied Mathematics, NYCU

March 9, 2022

Abstract

In this homework, we implement the deep learning method to classify the birds images. To avoid over-fitting, we use many augmentation to increase our training data. Then we train the model with five fold cross validation. For the test time, we collect the prediction which generate by five fold model and voting them. Code available at `https://github.com/Jia-Wei-Liao/CUB_200_2011_Dataset_Classification`.

1 Introduction

The Caltech-UCSD Birds-200-2011 (CUB-200-2011) dataset is the most widely-used dataset for fine-grained visual categorization task. It contains 11,788 images of 200 subcategories belonging to birds, 5,994 for training and 5,794 for testing. In this homework, we have 3,000 for training 3,300 for testing. So, the model is easy to overfitting.

2 Data Processing

2.1 Image Resize

Since the images in the dataset do not all have the same dimensions, we resize all images to suitable size. (375, 375) is the best size after we repeated attempts.

2.2 Random Transformation

To avoid over-fitting, we do the data augmentation by flip, rotation and affine transform with probability. On the other hand, We also attempt to increase the dataset by minor with left and right but it's not efficient so that we don't use.

2.3 Random Noise

To increase the generalization of our model, We add the noise with probability by $I(i,j)+\sigma \times s$ where $s \sim \mathcal{N}(0,1)$ and I is the color scale. Moreover, if we raise the probability of generating gaussian noise, the model's performance will decrease. Hence we set $p = 0.1$.

2.4 Image Normalize

the images are represented as tensors with pixel values ranging from 0 to 255. We apply z-score transform to the pixel values which is preferred by our model.

2.5 GridMask

GridMask is a simple, general, and efficient strategy. Given an input image, its algorithm randomly removes some pixels of it. We use four numbers $(r, d, \delta_x, \delta_y)$ to represent a mask.

- r is the ratio of the shorter gray edge in a unit.
- d is the length of one unit.
- δ_x and δ_y are the distances between the first intact unit and boundary of the image.

In fact, d is randomly chosen by our given interval and δ_x, δ_y are randomly generated by $[0, d]$.

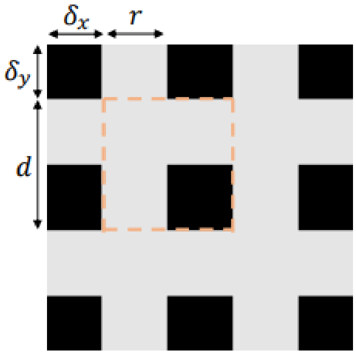


Figure 1. Mask

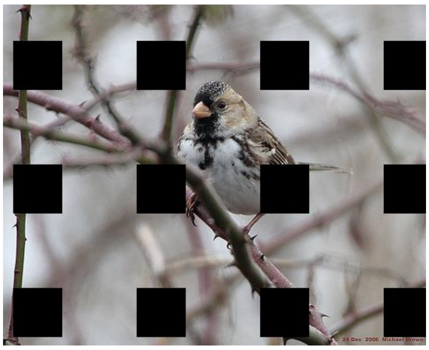


Figure 2. Example of GridMask

3 Training setup

3.1 Transfer Learning

First, we adopt the ResNet which is useful CNN-based model. In order to increase convergence rate, we use pretrained-weight on previous layers that is such a good initial. We also redefine the last layer to output 200 values, one for each class.

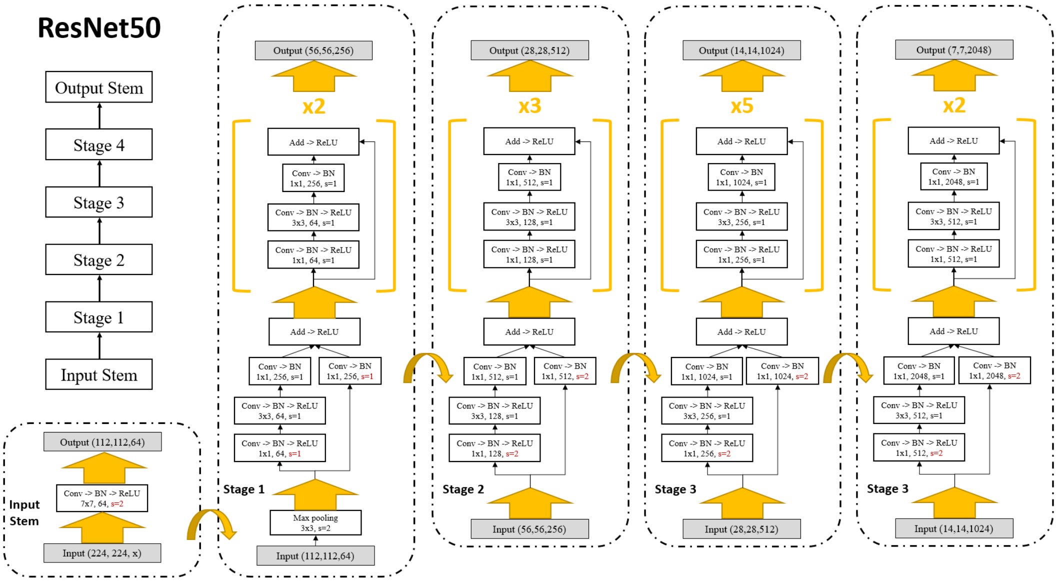
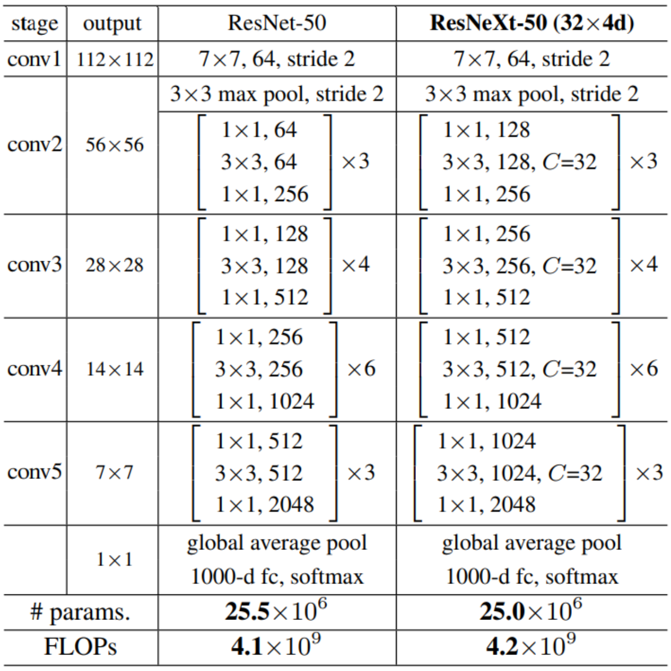

ResNet50
Output Stem
Stage 4
Stage 3
Stage 2
Stage 1
Input Stem
Output (112,112,64)
Conv -> BN -> ReLU 7x7, 64, s=2
Input Stem
Input (224, 224, x)
Output (56,56,256)
x2
Add -> ReLU
Conv -> BN 1x1, 256, s=1
Conv -> BN -> ReLU 3x3, 64, s=1
Conv -> BN -> ReLU 1x1, 64, s=1
Add -> ReLU
Conv -> BN 1x1, 256, s=1
Conv -> BN 1x1, 256, s=1
Conv -> BN -> ReLU 3x3, 64, s=1
Conv -> BN -> ReLU 1x1, 64, s=1
Stage 1
Max pooling 3x3, s=2
Input (112,112,64)
Output (28,28,512)
x3
Add -> ReLU
Conv -> BN 1x1, 512, s=1
Conv -> BN -> ReLU 3x3, 128, s=1
Conv -> BN -> ReLU 1x1, 128, s=1
Add -> ReLU
Conv -> BN 1x1, 512, s=1
Conv -> BN 1x1, 512, s=2
Conv -> BN -> ReLU 3x3, 128, s=1
Conv -> BN -> ReLU 1x1, 128, s=2
Stage 2
Input (56,56,256)
Output (14,14,1024)
x5
Add -> ReLU
Conv -> BN 1x1, 1024, s=1
Conv -> BN -> ReLU 3x3, 256, s=1
Conv -> BN -> ReLU 1x1, 256, s=1
Add -> ReLU
Conv -> BN 1x1, 1024, s=1
Conv -> BN 1x1, 1024, s=2
Conv -> BN -> ReLU 3x3, 256, s=1
Conv -> BN -> ReLU 1x1, 256, s=2
Stage 3
Input (28,28,512)
Output (7,7,2048)
x2
Add -> ReLU
Conv -> BN 1x1, 2048, s=1
Conv -> BN -> ReLU 3x3, 512, s=1
Conv -> BN -> ReLU 1x1, 512, s=1
Add -> ReLU
Conv -> BN 1x1, 2048, s=1
Conv -> BN 1x1, 2048, s=2
Conv -> BN -> ReLU 3x3, 512, s=1
Conv -> BN -> ReLU 1x1, 512, s=2
Stage 3
Input (14,14,1024)

Figure 3. ResNet50 framework (from Jia-Yau Shiau on the medium website)

After we pass the baseline, We also change the Network in order to increase the accuracy. Finally, we found a nice Network which is ResNeXt.



stage	output	ResNet-50	**ResNeXt-50 (32×4d)**
conv1	112×112	7×7, 64, stride 2	7×7, 64, stride 2
conv2	56×56	3×3 max pool, stride 2	3×3 max pool, stride 2
		$\begin{bmatrix} 1\times1, 64 \\ 3\times3, 64 \\ 1\times1, 256 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 128 \\ 3\times3, 128, C{=}32 \\ 1\times1, 256 \end{bmatrix}\times3$
conv3	28×28	$\begin{bmatrix} 1\times1, 128 \\ 3\times3, 128 \\ 1\times1, 512 \end{bmatrix}\times4$	$\begin{bmatrix} 1\times1, 256 \\ 3\times3, 256, C{=}32 \\ 1\times1, 512 \end{bmatrix}\times4$
conv4	14×14	$\begin{bmatrix} 1\times1, 256 \\ 3\times3, 256 \\ 1\times1, 1024 \end{bmatrix}\times6$	$\begin{bmatrix} 1\times1, 512 \\ 3\times3, 512, C{=}32 \\ 1\times1, 1024 \end{bmatrix}\times6$
conv5	7×7	$\begin{bmatrix} 1\times1, 512 \\ 3\times3, 512 \\ 1\times1, 2048 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 1024 \\ 3\times3, 1024, C{=}32 \\ 1\times1, 2048 \end{bmatrix}\times3$
	1×1	global average pool 1000-d fc, softmax	global average pool 1000-d fc, softmax
# params.		**25.5**$\times10^6$	**25.0**$\times10^6$
FLOPs		**4.1**$\times10^9$	**4.2**$\times10^9$

Figure 4. Compare ResNeXt-50 with ResNet50

3.2 Loss function

The category of our dataset is balance, so we use the cross entropy loss on this task. The cross entropy loss is defined by

$$\mathrm{CE}(\hat{y}, y) = -\sum_{i=1}^{200} y_i \log \hat{y}_i$$

where $y = (y_1, y_2, ..., y_{200})$ is one-hot label and $\hat{y} = (\hat{y}_1, \hat{y}_2, ..., \hat{y}_{200})$ is probability of our prediction.

3.3 Optimization

Adam is one of the most effective optimization algorithms for training but after our experiment, it seems ineffective on this task. So, We change to use AdamW. It is a stochastic optimization method that modifies the typical implementation of weight decay in Adam, by decoupling weight decay from the gradient update. We give the algorithm as the following:

Algorithm AdamW

```
Input: $\gamma, \beta_1, \beta_2, \theta_0, f(\theta), \varepsilon, \lambda$
Initial: $m_0 \leftarrow 0, v_0 \leftarrow 0,$
for $t = 1$ to ... do
    $g_t \leftarrow \nabla_\theta f_t(\theta_{t-1})$
    $\theta_t \leftarrow \theta_{t-1} - \gamma\lambda\theta_{t-1}$
    $m_t \leftarrow \beta_1 m_{t-1} + (1 - \beta_1) g_t$
    $v_t \leftarrow \beta_2 v_{t-1} + (1 - \beta_2) g_t^2$
    $\hat{m}_t \leftarrow \frac{m_t}{1 - \beta_1^t}$
    $\hat{v}_t \leftarrow \frac{v_t}{1 - \beta_2^t}$
end for
return $\theta_t$
```

3.4 Learning Rata Scheduling

To further improve results and make the model converge to the global minimum, we adjust the learning rate by exponent decay. It represent by the following:

$$\text{Learning Rate} = (\text{Initial Learning Rate}) \times (\text{Decay Factor})^{\frac{\text{Epoch}}{\text{Decay Period}}}$$

We set up the Initial Learning Rate $= 1e - 4,$ Decay Factor $= 0.8$ and Decay Period $= 3.$

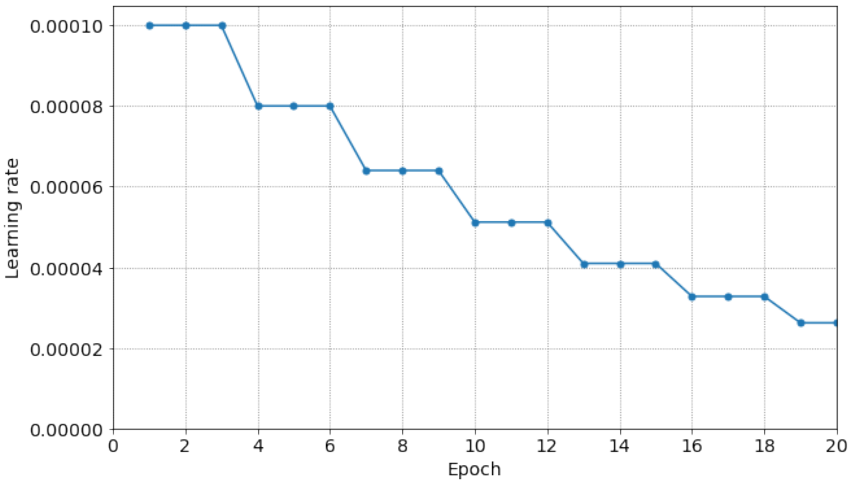


Figure 5. Learning rate at the different epoch

3.5 Five Fold Cross Validation

Cross Validation is a statistical method used to estimate the skill of machine learning models. It generally results in a less biased or less optimistic estimate of the model skill than other methods, such as a simple train and test split.

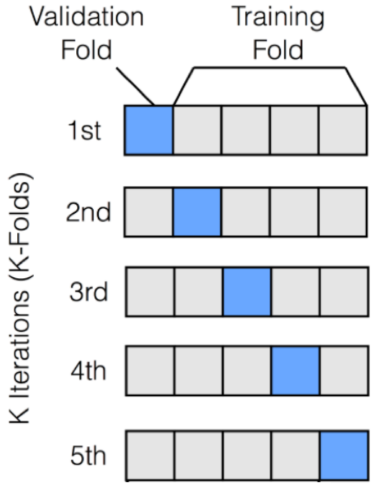


Figure 6. Schematic Diagram of *k*-fold cross validation

4 Ensembling

Ensemble methods are very powerful in improving the model's overall performance. So, we pick the best of 3 model for every fold to inference. For each image, We used 15 model to get the 15 prediction with probability and then we calculate the average as our final result.

5 Summary of Result

First, using ResNet50, we obtain a testing accuracy of 0.712826. Then using the five fold cross validation and ensemble technique, the testing accuracy goes up to 0.765249. we alse use ResNeXt-101, the testing accuracy rises to 0.793933. We choose the top 3 model of performance on the validation step to voting at the inference time. Finally, the testing accuracy improves to 0.811408.

The following table is the best accuracy of five fold cross validation.

Fold	1	2	3	4	5
Validation accuracy	0.8383	0.8267	0.8050	0.8250	0.8233

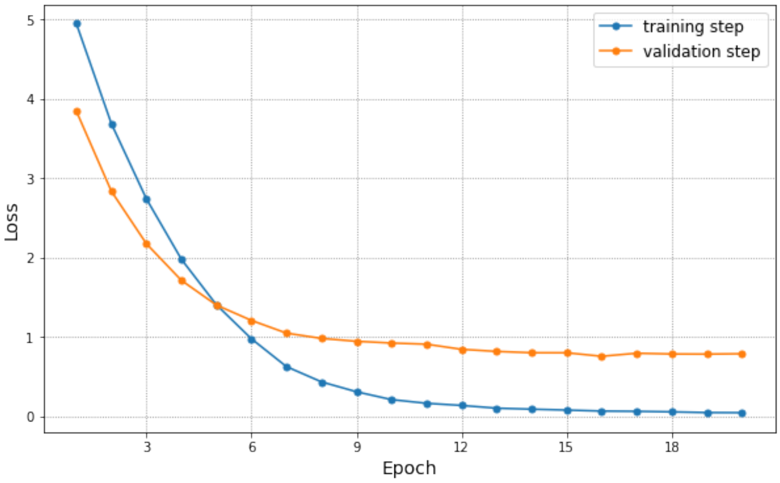


Figure 7. Loss Diagram for the fold 1

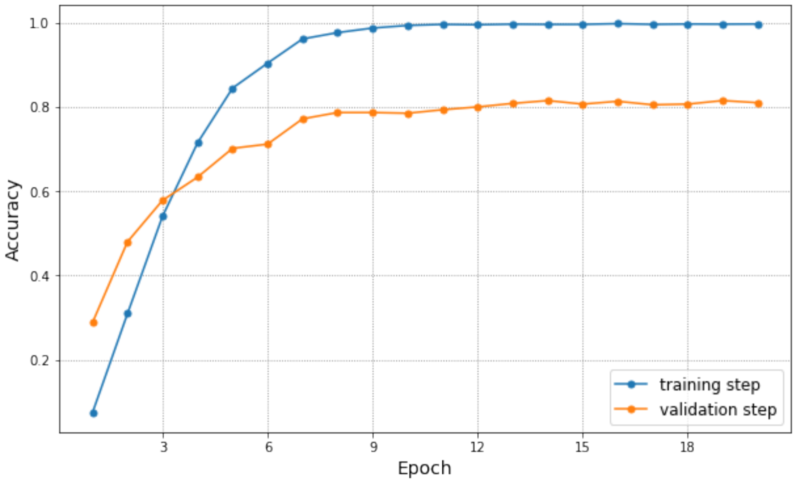


Figure 8. Accuracy Diagram for the fold 1

6 Feature Work

References

[1] Pengguang Chen, Shu Liu, Hengshuang Zhao, Jiaya Jia, GridMask Data Augmentation, *arXiv CVPR*, 2020.

[2] Kaiming He, Xiangyu Zhang, Shaoqing Ren, Jian Sun, Deep Residual Learning for Image Recognition, *CVPR*, 2015.

[3] Ilya Loshchilov, Frank Hutter, Decoupled Weight Decay Regularization, *ICLR*, 2019.

[4] Divya Shanmugam, Davis Blalock, Guha Balakrishnan, John Guttag, Better Aggregation in Test-Time Augmentation, *ICCV CVPR*, 2019.

[5] Saining Xie, Ross Girshick, Piotr Dollár, Zhuowen Tu, Kaiming He, Aggregated Residual Transformations for Deep Neural Networks *arXiv CVPR*, 2017.