# CNN Classification in CIFAR10

Bohao Zou, Jian Shi

June 7, 2020

## 1 Introduction

Image classification is the task of classifying an image into a class category. Many traditional models have been proposed to classify images such as KNN, SVM and Neural Network. But the result of those traditional models can't reach our goal and it is too hard to implement in real situation. A new model was proposed by Alex Krizhevsky at all at the time of 2012. This model is called CNN.

For exploring the best architecture of CNN model and pursing a breakthrough of the accuracy of image classification, we choose this problem that make a classification on the CIFAR-10 image data set. The basic hypothesis which we will make is that the distribution of training data set is as same as the distribution of testing data set in the CIFAR-10. The distinguish of the distribution of training set and testing set will have a huge impact on the performance of the CNN model.

In this project, we will train two special CNN models by using the data set CIFAR-10 and then using the Ensemble Learning Stacking to improve results. Stacking is an ensemble learning technique that combines multiple classification or regression models via a meta-classifier or a meta-regressor. The final accuracy of this project in this data set is 93.51%.

## 2 Methodology

### 2.1 CIFAR-10 Data Set

The CIFAR-10 data set consists of 60000 $32 \times 32$ colour images in 10 classes, with 6000 images per class. There are 50000 training images and 10000 test images. The data set is divided into training batches and one test batch. The test batch contains exactly 1000 randomly-selected images from each class. The training batches contain the remaining images in random order. The training batches contain exactly 5000 images from each class. The classes are completely mutually exclusive.

## 2.2 Mobile Net V2 [1]

Mobile Net V2 is one of the CNN model which used for mobile or the resource constrained environments. One of specify feature of this Net is that it has less parameters but the performance of this net in some data sets are excellent. This Net is established on the Inverted Residual Block. The Inverted Residual Block is improved from Depthwise Separable Convolutions.

The reason that choose this net is because we consider the efficiency of this net the high performance of this net in some cases.

### 2.2.1 Tiny Modify For Mobile Net V2

1. Replace the Batch Normalization[2] to Group Normalization[3] after the depthwise convolution. The intuition is that depthwise convolution performs lightweight filtering by applying a single convolutional filter for per input channel. The different convolutional filter will learn different information from per input channels. If we use Batch Normalization to the output of depthwise convolution, we would have a wrong estimator of the mean $\mu$ and variance $\sigma^2$ of each pixel in the output. This is because the per output channels of depthwise convolution comes from different convolutional filter, this may indicate that the distribution of each output channel is different.

2. From the paper [4] we can know that the performance of one net has highly connection with the resolution of input image, the depth of the net and the channels number of each layers. Inspired by this paper, we extend our Mobile Net V2 model in to $Original\_Depth \times 2$, $Original\_Channels \times 2$ and $Original\_Resolutions \times 1$.

## 2.3 ResNeSt Net[5]

ResNeSt Net is a modified convolution net from the ResNet[6]. In the paper, the authors added an Split-Attention block that enables attention across feature-map groups. The Split-Attention block is like the upgrade of the SE block[7] but enhance much nonlinear transformation ability compare with it. By stacking these Split-Attention blocks in ResNet-style, the authors obtain a new ResNet variant which we call ResNeSt.

The reason that why we choose this model is that it achieves the best performance in the classification on the Image-Net data set and the best mAP of the objection detection on the COCO data set. It is the SOTA model at present.

### 2.3.1 Tiny Modify For ResNeSt Net

1. Replace the Batch Normalization[2] to Group Normalization[3] in the split convolution blocks. The intuition is as same as the Mobile Net V2.

2. Inspired by this paper [4], we extend our ResNeSt Net (50 Layers) model in to $Original\_Depth \times 2$, $Original\_Channels \times 1$ and $Original\_Resolutions \times 1$.

### 2.4 Ensemble Learning, Stacking

In the algorithm Stacking, the base level models (Mobile Net V2, ResNeSt Net) are trained based on a complete training set, then we will build a meta-model which is trained on the outputs of the base level model as features and raise the result of accuracy.

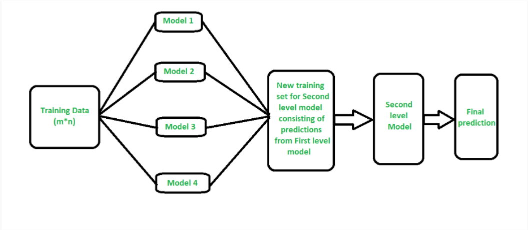
Training Data (m*n)
Model 1
Model 2
Model 3
Model 4
New training set for Second level model consisting of predictions from First level model
Second level Model
Final prediction

Figure 1: *The work flow of the algorithm Stacking.*

The meta-model which we used is a three layers neural network. There is no any activation in it. It is just make linear transformation for the output features of Mobile Net V2 and ResNeSt Net. For amplifying the effect of correct feature (i.e. The correct prediction output of one of net), we used 1024 hidden units for the output of Mobile Net V2 and ResNeSt Net. In the next step, we added the two linear transformed tensors and the predictive output is the linear transformation of the added tensor.

## 3 Implementation Details

### 3.1 Training Implementation

For fully training the Mobile Net V2 and ResNeSt Net, we trained those net 350 epoch for each net. The learning rate is also a key to train the net successfully. The learning rate schedule is: 0.1 for epoch $[0 \sim 150]$, 0.01 for epoch $[151 \sim 250]$ and 0.001 for epoch $[251 \sim 350]$. The mini-batch size is 128. The normalization layers are used after each convolution layers before the activation layers. Because the number of images in each class is same, so we used same weight Cross Entropy Loss. For enhancing the performance of each net, we used data augmentation for CIFAR-10 data set. The methods which we used are random crop and random horizontal flip. For guarantee fast convergence and higher precision of final result, we used Adam Bound [8] optimizer for optimizing. The feature of this optimizer is that it performs like Adam in the early stage of training and converts into SGD with momentum gradually. This feature gives it a fast convergence in the initial state and high precision result at the end of training.

For training meta-model, we used Cross Entropy Loss and SGD with momentum optimizer. The learning rate in this training procedure is a constant which is $1 \times 10^{-6}$. For fully training this meta-model, the epoch we set is 300.

All of those training processing is based on the GPU, GTX-1080Ti.

### 3.2 Regularization Implementation

Regularization is also a key to decide if we can train a excellent CNN model. The regular regularization method like dropout[9] is not fitful for CNN model. This is because dropout is fitful for fully connection layers. However, CNN is a partial connection layer. By solving this problem, we used Drop Connection [10] method. This regularization method can open or close one block with the *Bernoulli* distribution. In Mobile Net V2, this method is after Inverted Residual Block. In ResNeSt Net, this method is behind the ResNeSt block. The probability set for Mobile Net V2 is 0.45 and set 0.47 for ResNeSt Net. This is because ResNeSt is more complicate than Mobile Net V2. We used the dropout method at the last fully connection layer in those two CNN models. The probability for dropout is 0.75 and 0.77 for Mobile Net V2 and ResNeSt Net.

Beside using Drop Connection, we also apply the weight decay regularization. The value of this parameter is set $5 \times 10^{-4}$, $5.1 \times 10^{-4}$ and $1.5 \times 10^{-6}$ for Mobile Net V2, ResNeSt Net and meta-model respectively.

## 4 Results

The result of accuracy of those models are show below.

| | Accuracy |
|---|---|
| Mobile Net V2 | 92.58% |
| ResNeSt Net | 90.51% |
| Meta Model | 93.51% |

Table 1: *The accuracy of each trained models.*

Our assumption that the distribution of training data set is as same as the distribution of testing data set is true. This is because we get a 93.51% accuracy in the testing data set. If the distribution of training and testing are different, our model is hard to generalize performance from training data set to testing data set to get high accuracy.

From the table 1, we can know that the accuracy of complicate ResNeSt Net is lower than the simple Mobile Net V2. In general, a more complicate model will have more non-linear ability that have a greater result. However, in this case it is not. This may because of the resolution of image in this CIFAR-10 data set. For facing the low resolution image, a complicate CNN model have a greater probability to overfit than a simple CNN model. If we train a data set like Image Net with huge resolution, the ResNeSt may have better performance than the Mobile Net V2.

From the table 1, we can also see that the accuracy of Meta model is higher than any single CNN model. This indicates that Ensemble Learning, Stacking is useful when we have different types of models for one task. This algorithm will raise the performance about 1% $\sim$ 2%. In the future, if we face a problem which is hard to solve by one of machine leraning model, we can use Ensemble Learning to help us solving our difficult.

# References

[1] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, “Mobilenetv2: Inverted residuals and linear bottlenecks,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 4510–4520.

[2] S. Ioffe and C. Szegedy, “Batch normalization: Accelerating deep network training by reducing internal covariate shift,” *arXiv preprint arXiv:1502.03167*, 2015.

[3] Y. Wu and K. He, “Group normalization,” in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 3–19.

[4] M. Tan and Q. V. Le, “Efficientnet: Rethinking model scaling for convolutional neural networks,” *arXiv preprint arXiv:1905.11946*, 2019.

[5] H. Zhang, C. Wu, Z. Zhang, Y. Zhu, Z. Zhang, H. Lin, Y. Sun, T. He, J. Mueller, R. Manmatha *et al.*, “Resnest: Split-attention networks,” *arXiv preprint arXiv:2004.08955*, 2020.

[6] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[7] J. Hu, L. Shen, and G. Sun, “Squeeze-and-excitation networks,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 7132–7141.

[8] L. Luo, Y. Xiong, Y. Liu, and X. Sun, “Adaptive gradient methods with dynamic bound of learning rate,” *arXiv preprint arXiv:1902.09843*, 2019.

[9] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, “Dropout: a simple way to prevent neural networks from overfitting,” *The journal of machine learning research*, vol. 15, no. 1, pp. 1929–1958, 2014.

[10] L. Wan, M. Zeiler, S. Zhang, Y. Le Cun, and R. Fergus, “Regularization of neural networks using dropconnect,” in *International conference on machine learning*, 2013, pp. 1058–1066.