COSE474-2023F: Final Project
Mini-Multi-Class Classifier for Multi-Class Classification

2020170304 Wonjun Jang

1. Introduction

Multi-class classification is a machine learning task that involves classifying a sample into one of several classes. As one of the most common subdomains in computer vision, numerous models have been developed for this purpose, and there are many benchmark datasets available.

1.1. Motivation

Many real-world problems consist of multiple classes, and there are areas like healthcare where accurate class assignment becomes critical. In general, multi-class classification problems are less accurate than binary classification problems because it becomes more difficult to accurately learn and distinguish the boundaries between each class. This phenomenon becomes more pronounced as the number of classes that need to be distinguished increases, so finding a way to address it is the motivation for this research. Consequently, this study propose a novel model structure aimed at both reducing the complexity of distinguishing classes and enhancing accuracy through a mechanism that compensates for incorrect classifications.

1.2. Problem definition

This study defines the problem as follows. First, a model (Resnet-18) is selected to assess its performance in a multi-class classification problem with the aim of enhancing classification accuracy. Second, this study examines how the performance changes with the introduction of a new model structure and proposes an improved multi-classification model with detailed modifications. Third, in cases where performance does not show improvement, this study identifies weaknesses in the model structure and suggests directions for further research.

In summary, our contribution is a novel model structure designed to solve the performance degradation that arises with an increasing number of classes in a classification task.

2. Related Works

Among traditional machine learning techniques, SVMs are primarily applied to binary classification problems, but can be extended to multi-class classification problems(Hsu & Lin, 2002). SVMs solve multiple classification problems by breaking them down into binary classification problems in either an OvR or OvO fashion. In deep learning, due to the existence of the softmax function, it is possible to achieve good performance without using this approach in multi-classification problems. Recently studied Vit models(Dosovitskiy et al., 2020) can achieve 99.9% accuracy on some benchmark tasks. There is also research utilizing ensemble techniques for multi-class classification problems based on the RAKEL algorithm(Rokach et al., 2014). This is done by dividing the data classes into subgroups and training each classifier separately. Unlike previous approaches, our method addresses the problem by incorporating additional algorithms while preserving a simple model structure. this study explores the simultaneous training of multiple classifiers using the complete image-label pair dataset.

3. Method

The main challenge is to show how to decompose a large classification problem (with many classes) into smaller classification problems (with fewer classes) and how the smaller classifiers can help each other. If C of the classes to be classified are assigned to n classifiers, each classifier will perform classification on $[C/n]$ classes. The main idea here is to assign an extra class to each classifier. That is, the classifier will perform classification on $[C/n] + 1$ class. The output of each classifier is represented by f, where the subscript indicates the predicted class.

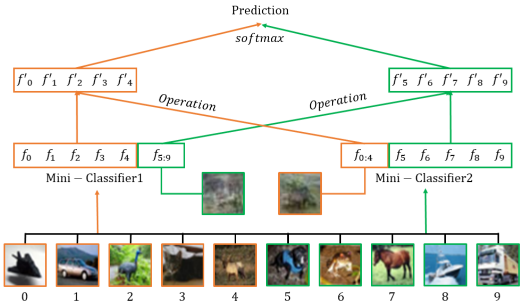
Prediction
$softmax$
f'_0 f'_1 f'_2 f'_3 f'_4
f'_5 f'_6 f'_7 f'_8 f'_9
$Operation$
$Operation$
f_0 f_1 f_2 f_3 f_4
$f_{5:9}$
$f_{0:4}$
f_5 f_6 f_7 f_8 f_9
Mini – Classifier1
Mini – Classifier2
0
1
2
3
4
5
6
7
8
9

Figure 1. The complete model structure formed by the combination of mini-classifiers.

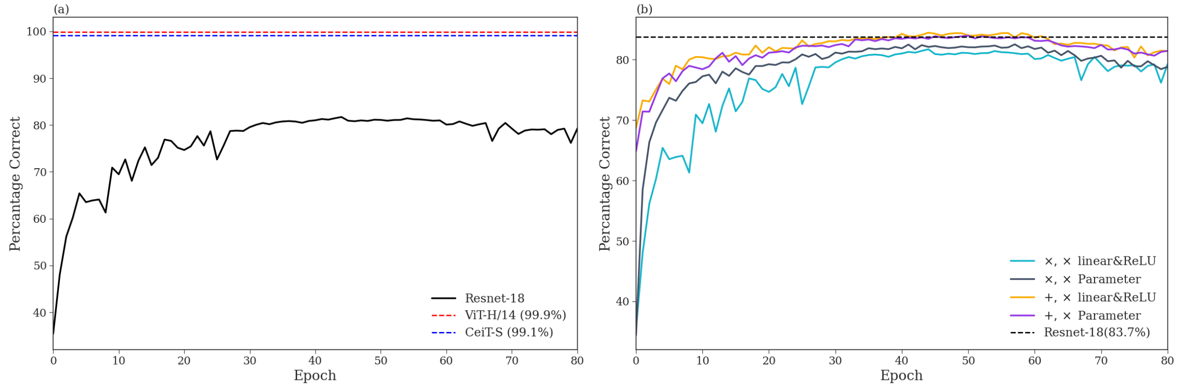

Figure 2. (a) Displayed in dotted lines are two state-of-the-art Vision Transformer (ViT) models, ViT-H/14 (1st, 99.9) and CeiT-S (13th, 99.1), while the solid lines represent the training log (valid score) of the pretrained ResNet-18. (b) Displayed in dotted line is the best valid score of the pretrained ResNet-18, while the solid lines represent the training log (valid score) of four methods.

If a classifier finds it difficult to classify some images into its assigned $[C/n]$ class, it should classify them as an extra class. These images may be assigned to a class associated with another classifier, and the extra class will capture the distinctive features of these images. Thus, for each image, if the prediction of each mini-classifier with the Extra Class is permitted to influence (add or multiply) the predictions of the other mini-classifiers for the $[C/n]$ classes, it can be expected to make more accurate predictions while having the effect of ensembling the predictions. This is analogous to having an additional classifier make decisions about challenging classes. The values predicted by extra class are only used to calibrate the normal class predictions, and the final prediction is made by concatenating the calibrated predictions from each classifier.

In mathematical terms, the process can be represented as follows:

$$f'_n = \text{softmax}(f_{in} \odot g(f_{j0})) \quad (i, j < n, i \neq j)$$

The predicted value of classifier i for class n is denoted by f_{in}. The $\odot$ means a special operation, which in this case is set to add or multiply. Taking the value at the 0th index from the output of the jth classifier, the model proceeds by passing it through the g function. This involves multiplying the result by a trainable parameter, or linear operation.

The operation of the model utilizing the given equation can be expressed in pseudo code as follows:

Algorithm 1 Combining Result Pseudo Code

```
$\text{Output } i, \text{Extra Class } i = f_i(x)[1:], f_i(x)[0]$
$\text{Output } j, \text{Extra Class } j = f_j(x)[1:], f_j(x)[0]$
$\text{Output } i = \text{Output } i \odot g(\text{Extra Class } j)$
$\text{Output } j = \text{Output } j \odot g(\text{Extra Class } i)$
$\text{Output} = \text{Concatenate}(\text{Output } i, \text{Output } j)$
```

The mathematical relationship governing the influence of one mini-classifier's Extra Class prediction on another mini-classifier's prediction is established through a series of experiments in the Experiments chapter.

4. Experiments

4.1. Dataset & Resource

The dataset employed for this study is CIFAR-10, a straightforward multi-class classification problem designed to serve as a benchmark for performance measurement. The dataset is divided into a training set and a test set, with 20% of the training set allocated for validation. The model is trained on the remaining 80% of the training set, and its performance is evaluated on the test set. The evaluation metric used is the Percentage Correct. The model is implemented using PyTorch and trained on a T4 GPU provided by Google Colaboratory.

4.2. Experimental setup

The mini-classifier utilizes ResNet-18, a model that, despite being relatively old, was chosen for its simplicity of implementation. This simplicity facilitates the introduction of new model structures, and the model's initially lower performance provides a clear basis for observing changes in performance.

The first step in our study involves finetuning a model pretrained on ImageNet using the ResNet-18 architecture on the CIFAR-10 dataset and assessing its performance. This initial performance will establish a baseline, and any improvement beyond this baseline score by the new model structure will indicate a performance enhancement.

The experimental variations in model structures are characterized by the computation method and the processing of extra class predictions. Two primary operation cases are considered: addition and multiplication. The handling of the predicted extra class values involves two steps: first, multiplication with a learnable parameter, and second, passage through linear and activation functions. In total, this study explore four distinct methods.

Quantitative results will be summarized in tables, and qualitative results will be visualized in graphs.

4.3. Results

Figure 2 (a) illustrates the baseline performance of the pre-trained ResNet-18, showcasing its best valid score of 83.7. This score serves as the reference point for comparison. In (b), the logarithmic representations of valid scores for the four models exhibit visual distinctions. Each experimental method is denoted as a pair ($\odot$, g). The corresponding maximum valid score and test score (at minimum valid loss) values are summarized in the table below.

Method	Valid Score	Test Score
Baseline	83.7	79.3
$(\times, \times\ linear\&ReLU)$	81.7	77.6
$(\times, \times\ Parameter)$	82.5	78.1
$(+, \times\ linear\&ReLU)$	**84.5**	79.2
$(+, \times\ Parameter)$	84.0	**79.4**

4.4. Discussion

The performance of the method employing the + operation is comparable to or exceeds the baseline performance in terms of both valid score and test score. This observation indicates that the novel model structure has played a role in enhancing the accuracy of predictions. However, merely surpassing the baseline does not necessarily imply a "good" model. Given the small performance difference, results might vary across different experimental environments. Therefore, it is crucial to conduct comparisons across multiple tasks to validate the observed performance difference.

The method utilizing the multiplication operation did not demonstrate favorable performance in any aspect comparable to the ways g did. Regarding the calibration of softmax values for the $[C/n]$ class across various mini-classifiers, this is attributed to the fact that multiplication can alter the case tendency of the softmax values themselves. Conversely, addition has the advantage of preserving that tendency. When images with n classes is input, each mini-classifier classifies it into $[C/n] + 1$ class. Consequently, each classification may lack clearly distinguishable features. The outcomes appear less distinct than anticipated, likely due to the ambiguous classification into 6 classes rather than the intended clear distinction among 10 classes.

5. Future Works

Based on the validation data, the novel model structure proposed in this paper marginally surpasses the performance of the existing ResNet-18. However, as we did not observe a significant improvement in performance based on the test data, there is potential for future exploration in the following directions.

First, it is worth considering experiments involving tasks with a larger number of classes. The current dataset has a relatively small number of classes due to our experimentation with CIFAR-10. Given the class imbalance in the test data, experiments assessed performance using the mean percentage correct, and subtle improvements are not easily discernible in the numerical results. Therefore, this difference may be more noticeable if the model structure is tested on more difficult tasks with more classes.

Second, consider using a lightweight model as one mini-classifier. Currently, we employ two ResNet-18 models as mini-classifiers, which doubles the number of parameters to be trained. If the performance gain from introducing a new structure is outweighed by the cost of doubling the parameters, its practical use becomes challenging. Currently, mini-classifiers play an equivalent role and support each other. Alternatively, it's feasible to have one model for primary classification and another for sub-classification, with the sub-model assisting the main model. By using a simpler structure such as MLP for the sub-model, the number of parameters to be learned can be significantly reduced while still serving the purpose of aiding in classification, thereby reducing costs.

6. Conclusion

In this paper, a novel model structure is introduced to address the performance degradation observed in multi-class classification tasks as the number of classes increases. The proposed model decomposes a complex classification problem into smaller subproblems, assigning each subproblem to a mini-classifier. These mini-classifiers collaborate to handle challenging classifications by incorporating additional class predictions. The Extra Class predictions are then utilized to calibrate the standard class predictions, thereby improving the overall accuracy.

Shown in the experiments is that the proposed model structure slightly outperforms the default ResNet-18 model; however, the performance improvement is less pronounced, suggesting that further exploration is warranted. Future work could therefore investigate the effectiveness of the model on tasks with a large number of classes to see larger performance changes, and consider using a lightweight model for one mini-classifier.

References

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

Hsu, C.-W. and Lin, C.-J. A comparison of methods for multiclass support vector machines. *IEEE Transactions on Neural Networks*, 13(2):415–425, 2002. doi: 10.1109/72.991427.

Rokach, L., Schclar, A., and Itach, E. Ensemble methods for multi-label classification, 2013.

Yuan, K., Guo, S., Liu, Z., Zhou, A., Yu, F., and Wu, W. Incorporating convolution designs into visual transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 579–588, 2021.

(Dosovitskiy et al., 2020) (Hsu & Lin, 2002) (Yuan et al., 2021) (Rokach et al., 2013)