Long-Tail Class Incremental Learning via Independent Sub-prototype Construction

Supplementary Material

1. Hyperparameters sensitivity analysis

In this section, we provide a more detailed exposition of the hyperparameters utilized in the paper.

1.1. The number of independent sub-prototype basis vectors

In the experiment, the number of independent sub-prototype basis vectors was set to twice the number of classes; for example, when the task setting is $50 + 10 * 5$, the number of independent sub-prototype basis vectors is 100 in the first task, and gradually increases in steps of 20 to 200 as the number of tasks increases. The selection of the number of sub-prototype basis vectors was determined through experimentation. We exclusively examined the relationship between the classification accuracy of the model obtained on the first task and the number of basis vectors, the imbalanced rate $\rho = 0.01$ in CIFAR100, Shuffled LT-CIL, and the results are shown in Fig. 1.

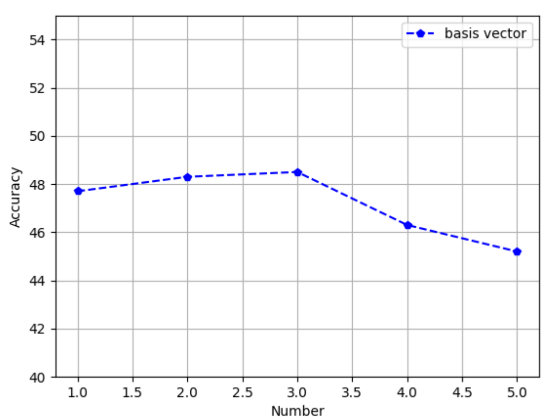

Figure 1. The relationship between the number of sub-prototype basis vectors and classification accuracy.

The x-axis represents the ratio of the number of basis vectors to the total number of classes. The experimental results show that an increase in the number of basis vectors does not necessarily lead to improved performance. Our analysis suggests that this may be attributed to the excessive dispersion of learning among sub-prototypes. Therefore, in our method, we chose the number of sub-prototypes to be twice the total number of classes.

1.2. The weight of construction loss

To address the issue of intra-task imbalance, we propose the sub-prototype space. By leveraging the features re-sampled within the sub-prototype space, we aim to mitigate the impact of data imbalance on the model. Regarding the hyperparameters before the construction loss λ_1, we conducted experiments under $\rho = 0.01$ on CIFAR100, Shuffled LT-CIL, and 5 tasks. The experimental results are depicted in Fig. 2.

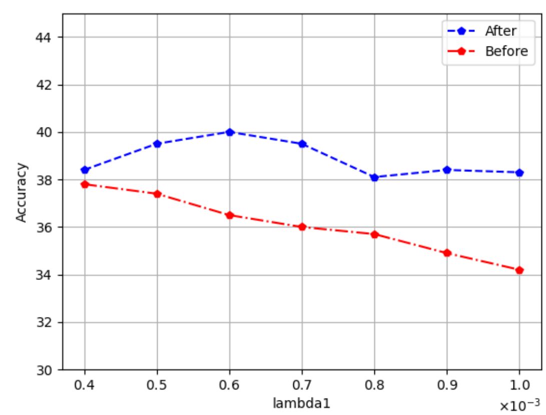

Figure 2. Sensitivity analysis of hyperparameters λ_1 .

In the experimental results, "before" and "after" signify the stages where only the sub-prototype space is constructed, and re-sampled features are utilized to adjust the model. It can be seen that only the construction phase affects the model during the gradual increase of the parameters, resulting in a slight decrease in the performance of the model, but when the model is fine-tuned using the re-sampled features, the performance of the model obtains a great enhancement, but this enhancement does not continue to rise as the parameters increase. Since the construction of the sub-prototype space also depends on the performance of the model, a balance needs to be found, so the value is taken in the method: $\lambda_1 = 6 \times 10^{-4}$.

1.3. The weight of distillation loss1

In order to prevent the collapse of the sub-prototype space by new knowledge, we propose the reminiscence space to constrain the sub-prototype space and propose a distillation loss of the sub-prototype space $\mathcal{L}_{dis1}$. The experimental setup is the same as before, and the experiment results are

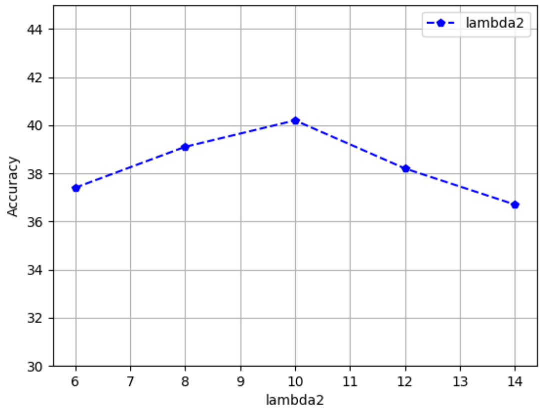

Figure 3. Sensitivity analysis of hyperparameters λ_2 .

shown in Fig. 3. The distillation loss too small can lead to insufficient constraints on the sub-prototype space and forgetting too much new knowledge, and too large can lead to too strong constraints on the model to learn new knowledge, both of which can lead to a decrease in the overall performance of the model, and therefore in our experiments we choose $\lambda_2 = 10$

1.4. The weight of distillation loss2

We also use the features sampled in the reminiscence space to train the model to prevent the model from forgetting previously learned knowledge, corresponding to the distillation loss $\mathcal{L}_{dis2}$. The experimental setup is the same as before, and the experiment results are shown in Fig. 4. The final

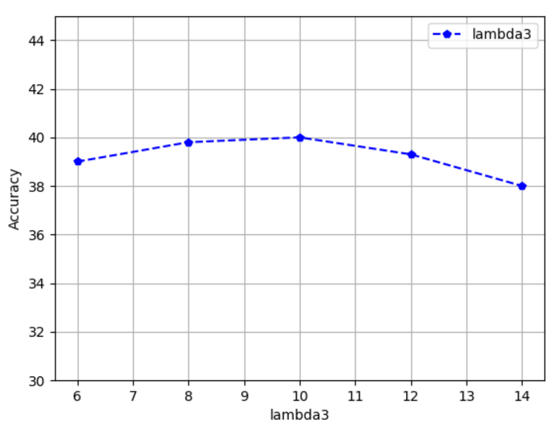

Figure 4. Sensitivity analysis of hyperparameters λ_3 .

setting in our method is:$\lambda_3 = 10$.

2. Re-sampled features ablation study

In section 3.2.2, we described our method of re-sampling features using the sub-prototype space, and to make our experiments more convincing, we remove the sub-prototype space and only do the same number of feature replications in the same setting as the similarity matrix, experiments on task 0 are shown in Figure 5. And a) is the results of removing the sub-prototype space, and b) is our method. The examples chosen are samples from the minority class, and it is clear that re-sampling the samples from the sub-prototype space solves the problem caused by imbalanced data distribution, allowing the model to learn more specific information about the minority classes.

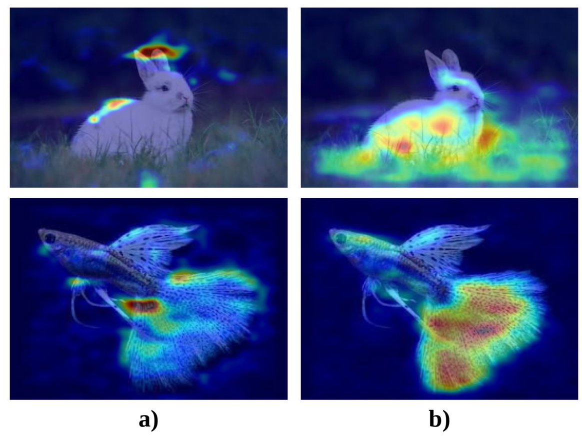

Figure 5. Re-sampled features ablation study. The setting is the same as the similarity matrix, and b) is our method and a) is without sub-prototype place.