Incremental Learning Using Conditional Adversarial Networks

Ye Xiang[1] Ying Fu[1] Pan Ji[2] Hua Huang[1]
[1]Beijing Institute of Technology [2]NEC Labs America
{xiangye,fuying,huahuang}@bit.edu.cn, peterji1990@gmail.com

1. Experimental Results

In this supplementary material, we first discuss the number of convolution layers for G. Secondly, we compare the effect of full covariance matrix and diagonal covariance matrix. Thirdly, we conduct an another evaluation on ImageNet dataset. Fourthly, we experiment with a different setup. Finally, we validate the robustness of our method to the change of underlying CNN architectures.

1.1. The Number of Convolution Layers for G

In this experiment, we discuss the number of convolution layers for G. Different values including 3, 4, and 5 are respectively used and compared on public CIFAR-100 dataset with the incremental step value as 10 classes. The results are shown in Table 1. We can see that the best α_{all} is achieved by 4 convolution layers, and a quite similar result is reached by 3 convolution layers. Hence, considering the computational complexity, we decide to set the number of convolution layers for G to be 3.

Table 1. Comparison of different numbers for convolution layers of G on CIFAR-100 dataset with the incremental step value as 10 classes.

$\#$ *conv layers*	3	4	5
α_{orig}	0.644	0.649	0.642
α_{new}	0.830	0.821	0.825
α_{all}	0.671	**0.673**	0.668

1.2. Full Covariance v.s. Diagonal Covariance

Within our storage for old classes, the covariance matrix consumes the most memory. In view of this, we attempt to explore the effect of diagonal covariance matrix which is more memory-efficient. The comparison result with the original full covariance matrix is shown in Table 2, for which the CIFAR-100 dataset with the incremental step value as 10 classes is used. We can see that the diagonal covariance matrix lacks in keeping large amount of useful information for recognizing old classes, and thus performs significantly worse directing at α_{orig} and α_{all}. Hence we stick to utilizing the full covariance matrix.

Table 2. Comparison between full covariance matrix and diagonal covariance matrix on CIFAR-100 dataset with the incremental step value as 10 classes.

	Full Covariance	Diagonal Covariance
α_{orig}	0.644	0.579
α_{new}	0.830	0.845
α_{all}	0.671	0.609

1.3. Evaluation on ImageNet

We follow the setup in FearNet and randomly select half of the classes (500 classes) to train an initial model. The remaining classes are uniformly added by 100 classes per incremental step. Following iCaRL and End-to-End, we report the top-5 accuracies. As shown in Table 3, our method achieves the best accuracies for incremental learning on ImageNet.

Table 3. Comparison of top-5 test accuracies on ImageNet.

	iCaRL	End-to-End	FearNet	Ours
α_{orig}	0.407	0.384	0.450	0.475
α_{new}	0.612	0.605	0.623	0.637
α_{all}	0.456	0.428	0.493	**0.509**

1.4. Different Experimental Setup

Here we try a different experimental setup, i.e., splitting the whole dataset into batches with the same number of classes (20 classes), as in iCaRL and End-to-End. The result on CIFAR-100 is shown in Table 4. It can be seen that our proposed method still achieves the STOA performance.

Table 4. Comparison of test accuracies on CIFAR-100 with a different experimental setup.

	iCaRL	End-to-End	FearNet	Ours
α_{orig}	0.658	0.639	0.649	0.661
α_{new}	0.654	0.645	0.691	0.719
α_{all}	0.631	0.614	0.668	**0.676**

1.5. Different Underlying CNN Architectures

Among the STOA baseline methods, iCaRL and End-to-End take ResNet-32 as the backbone network, while Fear-

Net and ours use ResNet-50. Here for fair comparison, we take ResNet-32 for all methods (including ours), and evaluate them on the CIFAR-100 dataset. The results are shown in Table 5. We can conclude that our method is robust to the change of underlying CNN architectures, and the peformance improvement is truly due to the proposed incremental learning strategy.

Table 5. Comparison of test accuracies for all methods with ResNet-32.

	iCaRL	End-to-End	FearNet	Ours
α_{orig}	0.569	0.507	0.640	0.645
α_{new}	0.673	0.696	0.681	0.708
α_{all}	0.609	0.567	0.652	**0.663**