5C-01

Continual Learning for Deep Neural Networks

Jiyuan Sun Lei Ma Jianjun Zhao
Graduate School and Faculty of Information Science and Electrical Engineering,
Kyushu University

Abstract

Deep Neural Networks (DNNs) have shown promising performance in various kinds of applications. However, different from human brain, a well-trained DNN is not capable of remembering old classes when learning new classes, which is called catastrophic forgetting of neural networks.

In this work, one approach based on feature extraction for DNNs to overcome the forgetting problem is proposed. It tries to maintain the knowledge of old classes by building and storing average feature vectors of the training data seen so far. Testing with MNIST and CIFAR dataset, we prove that our approach can efficiently decrease the cost for DNNs' continual learning of new classes as there is no need to retrain all the old classes.

1. Introduction and Background

In recent years, deep neural networks (DNNs) have achieved great success in various kinds of fields,such as computer vision and natural language processing. However, most DNNs could not learn new tasks without forgetting previously learned tasks, which is called catastrophic forgetting.

Our goal is to develop a strategy of continual learning, which enables DNNs to learn new knowledge without deterioration in the performance of old tasks. In this paper, we propose a method based on feature extraction and distilling technique in DNNs.The continual learning method could be tested with MNIST and CIFAR dataset. The original dataset and the continual learning result of the network could be visualized by scatter plotting, which helps to evaluate learning performance.

2. Related Works

LWT[1] is a technique that combines finetuning and distilling networks[2] and doesn't need the access to the old dataset. When new classes come,new nodes are added and then trained with much smaller learning rate. iCaRL[3] solves the forgetting problem by extracting high level features with DNNs of each class and then outputs the result with K nearest neighbor classifier by calculating the distance between the input with features. DeepMind of Google proposes EWC (Elastic Weight Consolidation) [4], which, by adding Fisher loss in the training loss function, tries to constrain the training adjustment of DNNs parameters in a region of low error rate for both old and new classes.

There are also other kinds of techniques developed to solve the forgetting problem, such as changing the structure or activation function of DNNs, using GAN to generate virtual historical data for retraining [5].

3. Method

We try to solve the catastrophic forgetting by extracting the features learned by DNNs and train new tasks with distillation loss, whose teacher network is the network trained with old dataset. Algorithm1 shows the whole process of two techniques used.

3.1 Feature Extraction

We use the neural network to extract the features of the dataset. The features of each class are extracted as vectors. With these vectors, some representative data that are the closest to these features are selected and retrained together with the training data of new classes. Combining training dataset of new classes with the sampled data from the previously trained class, we believe the network could remember more old knowledge.

3.2 Distillation Training

We also apply the distilling[2] technique in our method. The idea of distilling is to use a teacher network to guide the training of a simpler student network by adding a distillation loss.

$$L(\theta, y1, y2, y\_lab) = \lambda L_d(y1, y\_lab) + L_s(y2, y\_lab) + R(\theta) \quad (1)$$

The cross entropy loss function for the

continual learning is shown in formula (1) up, where L_s is the standard loss, which is the cross entropy between output and labels, and L_d is the distillation loss, which encourages the current training to output the same result with the previously trained network. R is regularization to reduce the complexity of models to avoid overfitting. λ is a coefficient which determines the importance of L_d.

Algorithm1.code for the proposed method

```
Input   x1,x2.....xn   # training data of each class
For i=1,2,....n do
    if   i=1 then
        # Network training with loss function
      Network_train(xi,loss=Ls+R)
    else
      yo=Teacher_Network(xi)
        # training with distillation loss
      Network_train(xi,sample,loss=Ld+Ls+R)
    end if
    feature_vec=extraction(network_output)
    SampleSet_SelectAdd(xi,feature_vec)
      # save weight and bias for teacher network
    SaveParameter(Network)
```

We also use t-SNE[6] to visualize the original data and the features after training. t-SNE converts high-dimensional euclidean distances between two datapoints into the conditional possibility that one picks another as its neighbour. With t-SNE, we can visualize and evaluate the quality of the features learned by the network directly and quickly.

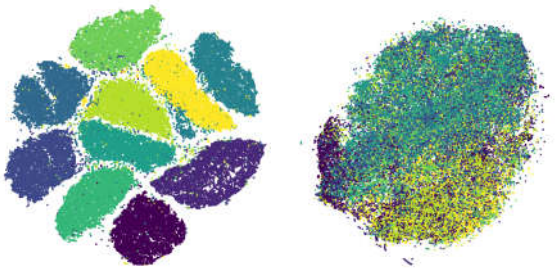
4. Experiment and Future work

Our method would be implemented with tensorflow as it's currently the most famous deep learning framework, with which it becomes very easy to build DNNs, obtain and store the output of each layer. The method would be tested with MNIST and Cifar10, both of which have 10 classes of images. We choose LeNet5 for MNIST and ResNet for CIFAR as the neural network models. Table1 is the information of MNIST and Cifar10, such as the number of classes and training images and the size of images.

Table1.details of the two dataset

Dataset	Class	Train	Test	Size
MNIST	10	60,000	10,000	28*28*1
CIFAR10	10	50,000	10,000	32*32*3

The visualizing result of the original training and testing set of these two datasets is displayed in Fig1. It can be seen that different classes in the dataset are not clearly separated, especially for the CIFAR10. Features learned by the network trained by proposed continual learning method could also be visualized to evaluate its performance. It can be expected that output features of the network could be clearly distinguished as 10 groups after the network is trained with the method proposed.



a).MNIST **b).CIFAR10**

Fig.1 t-SNE result of Dataset

Reference:

[1] Li, Zhizhong, and Derek Hoiem. "Learning without forgetting." IEEE Transactions on Pattern Analysis and Machine Intelligence 40.12 (2018): 2935-2947.

[2] Hinton, Geoffrey, Oriol Vinyals, and Jeff Dean. "Distilling the knowledge in a neural network." arXiv preprint arXiv:1503.02531 (2015).

[3] Rebuffi, Sylvestre-Alvise, et al. "icarl: Incremental classifier and representation learning." Proc. CVPR. 2017.

[4] Kirkpatrick, James, et al. "Overcoming catastrophic forgetting in neural networks." Proceedings of the national academy of sciences (2017): 201611835.

[5] Wu, Yue, et al. "Incremental Classifier Learning with Generative Adversarial Networks." arXiv preprint arXiv:1802.00853 (2018).

[6] Maaten, Laurens van der, and Geoffrey Hinton. "Visualizing data using t-SNE." Journal of machine learning research 9.Nov (2008): 2579-2605.