RE-IMPLEMENT OF DATASET CONDENSATION WITH GRADIENT MATCHING

Qidong Fan, Xiangru Chen & Yuqi Yi
School of Electronics and Computer Science
Southampton University
Southampton, UK
{qf1u20, xc3e20, yy12g20}@soton.ac.uk

ABSTRACT

Larger data sets in many fields have become a part of training machine learning models. However, these large data sets usually require relatively more time and cost for storage and preprocessing. In response to this problem, Bo Zhao et al. (2021). proposed a data synthesis technology for compressed data sets. This report is a re-implementation of the gradient matching-based dataset compression technology of Bo Zhao et al.

1 INTRODUCTION

The most advanced machine learning models usually represent the need to process large-scale data sets containing millions of samples. Large-scale data sets require more resources both in storage and preprocessing. Therefore, the data selection method improves the data efficiency of machine learning by identifying the most representative data samples. Because the classic data selection method coreset constructionAgarwal et al. (2004) method has shortcomings. Bo Zhao et al. were inspired by the dataset distillation(DD)Wang et al. (2018) method to try this method of synthesizing new samples to condense the big data set.
The original paper author trained and evaluated the classification performance of condensed images on the four standard benchmark data sets of MNISTLecun et al. (1998), SVNHNetzer et al. (2011), FashionMNISTXiao et al. (2017), and CIFAR10Krizhevsky et al. (2009) by using six standard deep network architectures. Our team used the GPU provided by the ECS to reproduce the main code provided by the author of the original paper on Github.
In the gradient descent matching, the core code of data set condensation, we use the original author's code. To verify different data sets using different network models by adding our own new methods.

2 EXPERIMENT

2.1 EXPERIMENTAL SETUP

All experiments were run on the Southampton ECS GPU compute platform with single RTX2070s and Google colab with GPU support in a Jupyter environment. The requirements packages are numpy=1.19.1, scipy=1.5.0, torch=1.6.0, torchvision=0.7.0, pillow=7.2.0, cudnn= 7.6.5, python=3.8.
In the following experiments, the hyperparams specified by the original author were used.

2.2 DATASET AND ALGORITHM

First, we import four data sets that are widely used in evaluating the classification performance of condensed images. MNIST, a large database of handwritten digits. Fashion MNIDT, a clothing classification of Zalando's article images. SVHN, a real world digits image database. CIFAR19, a object recognition 10 classes database. At the same time, we built different models to train and test the condensed dataset. These data models are MLP, ConvNet, AlexNet, VGG and ResNet.

Algorithm 1 Dataset condensation by using gradient matching

Input: Training image$\mathcal{T}$;
Output: S

Required: Randomly initialized set of synthetic samples $\mathcal{S}$ for C classes, probability distribution over randomly initialized weights $P_{\boldsymbol{\theta}_0}$, deep neural network $\phi_{\boldsymbol{\theta}}$, number of outer-loop steps K, number of inner-loop steps T, number of steps for updating weights $\varsigma_{\boldsymbol{\theta}}$ and synthetic samples $\varsigma_{\mathcal{S}}$ in each inner-loop step respectively, learning rates for updating weights $\eta_{\boldsymbol{\theta}}$ and synthetic samples $\eta_{\mathcal{S}}$.
for $k = 0, \cdots, K-1$ **do**
  Initialize $\boldsymbol{\theta}_0 \sim P_{\boldsymbol{\theta}_0}$
  for $t = 0, \cdots, T-1$ **do**
    for $c = 0, \cdots, C-1$ **do**
      Sample a minibatch pair $B_c^{\mathcal{T}} \sim \mathcal{T}$ and $B_c^{\mathcal{S}} \sim \mathcal{S}$ ▷ $B_c^{\mathcal{T}}$ and $B_c^{\mathcal{S}}$ are of the same class c.
      Compute $\mathcal{L}_c^{\mathcal{T}} = \frac{1}{|B_c^{\mathcal{T}}|} \sum_{(\boldsymbol{x},y)\in B_c^{\mathcal{T}}} \ell(\phi_{\boldsymbol{\theta}_t}(\boldsymbol{x}), y)$ and $\mathcal{L}_c^{\mathcal{S}} = \frac{1}{|B_c^{\mathcal{S}}|} \sum_{(\boldsymbol{s},y)\in B_c^{\mathcal{S}}} \ell(\phi_{\boldsymbol{\theta}_t}(\boldsymbol{s}), y)$
      Update $\mathcal{S}_c \leftarrow \texttt{opt-alg}_{\mathcal{S}}(D(\nabla_{\boldsymbol{\theta}}\mathcal{L}_c^{\mathcal{S}}(\boldsymbol{\theta}_t), \nabla_{\boldsymbol{\theta}}\mathcal{L}_c^{\mathcal{T}}(\boldsymbol{\theta}_t)), \varsigma_{\mathcal{S}}, \eta_{\mathcal{S}})$
    Update $\boldsymbol{\theta}_{t+1} \leftarrow \texttt{opt-alg}_{\boldsymbol{\theta}}(\mathcal{L}^{\mathcal{S}}(\boldsymbol{\theta}_t), \varsigma_{\boldsymbol{\theta}}, \eta_{\boldsymbol{\theta}})$ ▷ Use the whole $\mathcal{S}$

2.3 TRANING RESULTS

Table 1: Compared with the results of the original paper and other Coreset methods. Img/Cls is the number of images extracted from each class, ratio is the proportion, GradientM is the accuracy obtained in the original paper, and Our is the result obtained by our team.

	Img/Cls	Ratio	Random	GradientM	Our	Whole Dataset
MNIST	1	0.017	62.3	91.7 ± 0.5	**91.94**	99.5
	10	0.17	94.4	97.4 ±0.2	**97.25**	
	15	0.25	96.2	-	**97.78**	
FashionMNIST	1	0.017	52.5	70.5 ±0.6	**70.47**	93.5
	10	0.17	73.9	82.3 ±0.4	**82.55**	
	15	0.25	77.2	-	**82.68**	
SVHN	1	0.014	14.6	31.2 ±1.4	**31.15**	95.3
	10	0.14	32.7	76.1 ±0.6	**76.12**	
	15	0.21	56.9	-	**78.2**	
CIFAR10	1	0.02	14.6	28.3 ±0.5	**28.09**	84.2
	10	0.2	25	44.9 ±0.5	**45.23**	
	15	0.3	44.3	-	**47.64**	

First compare the reproductive method with the traditional data condensed method, the choices of coreset selection are Random, Herding, K-Center, Forgetting by using ConvNet model to train and test the four datasets. The number of images in each classes are 1 and 10 for training and testing. When the number of classified pictures is 10, the training time of a single dataset has exceeded 6 hours by using gtx2070 graphics card. Therefore, due to time constraints, the image per class, 20,30,40,50 of the original paper is not realized.original paper was not included.

Data set compression consists of two stages. The first step is to learn the feature selection of compressed images, and the second step is to use compressed images for classification training. Due to the limitation of computational performance, the parameter settings of the original paper were not kept consistent, and the parameters were set to 1, 10, 15.

Each method was run 5 times, and 20 models were randomly initialized in each run, and all models were evaluated at the end. The data model evaluation results are roughly consistent with the original paper. The data results show that the condensed dataset obtained by gradient matching can maintain high accuracy even when the compression rate is high. For example, the MNIST dataset has a compression rate of 83% and an accuracy of 97.25%. Is close to 99.5% accuracy of the whole training dataset in Table 1.

Figure 1: Visualization of four datasets for MNIST,FashionMNIST,SVHN,CIFAR10

Figure 1 is the data compression set of each classification obtained by the method of the original paper. We can see that these compressed data sets are very representative. It shows that the method is feasible from a macro point of view.

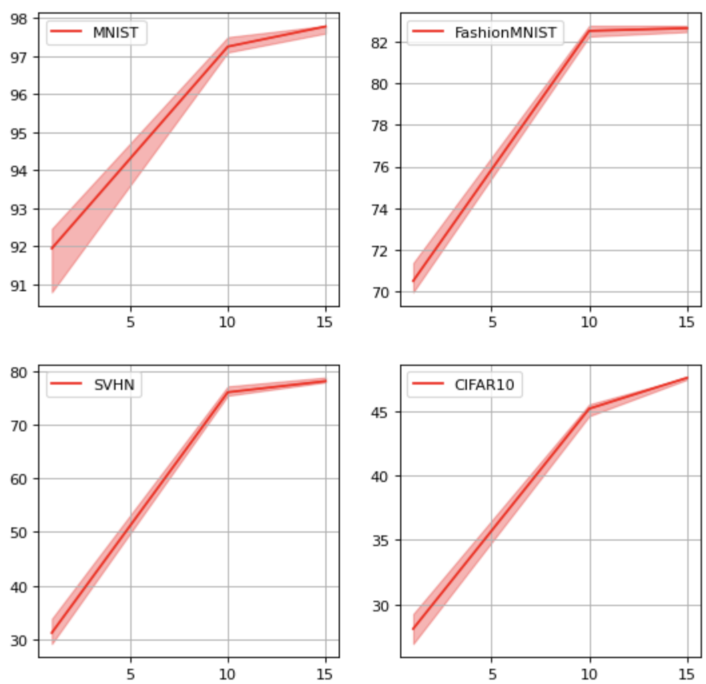

Figure 2: Accuracy of data set classification

In Figure 2, is the classifiaction of accurate with using the condensed data set of four whole data sets. Samples were taken at img/cls = 1,10 and 15, respectively. Shadow is the accuracy range in different epoch. The original paper is learning 1, 10 and 50 images/class on four standard benchmark data sets. However, due to current computing performance limitations, image class=50 is far beyond the acceptable range. The team made a change from 50 images/class to 15 images/class. By running each method 5 times, 5 composite sets and 20 random initialization models for each composite set are obtained.

In Table2, we can see that although the original paper method is inspired by DD method, the accuracy of the condensed data set for training with ConvNet used is still very different. In MNIST and Img/Cls=10, the accuracy of DD method is 74.3%, the result of original paper will get 93%. Our team got the same result as the original paper through training.

Table 2: The test accuracy compared with DD method and result of the original paper and our team.

Dataset	Img/Cls	DD	GradientM	Our
MNIST	1	-	85	86.13
	10	74.3	93	93.97
CIFAR10	1	-	24.2	24.68
	10	37.2	39.1	38.7

3 Conclusion

Firstly, the method and data of this paper are realizable. Our group basically completed the reproduction of the original paper data through verification. By writing some new codes, such as different training models, to further demonstrate the literary theory. Due to the limitation of computing equipment, part of the data can not be reproduced, because the training time is too long.
The synthetic sample is generated by matching the model gradient with the model gradient obtained on the original input image. After training, it is proved that these compressed image sets are model-independent, so they can be used to train different neural models. Compared with the existing methods, the proposed method is efficient and effective, and achieves the most advanced performance. It can also be used to train any other networks with different architectures, which makes the method more applicable. The method has been used in MNIST, SVHN and CIFAR10, etc. It has been verified on several smaller data sets.

4 Appendix

The reimplement code, data and experiment report can be obtained in GitHub repositories:

References

Pankaj K. Agarwal, Sariel Har-Peled, and Kasturi R. Varadarajan. Approximating extent measures of points. *J. ACM*, 51(4):606–635, July 2004. ISSN 0004-5411. doi: 10.1145/1008731.1008736. URL `https://doi.org/10.1145/1008731.1008736`.

Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.

Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998. doi: 10.1109/5.726791.

Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Bo Wu, and Andrew Y. Ng. Reading digits in natural images with unsupervised feature learning. In *NIPS Workshop on Deep Learning and Unsupervised Feature Learning 2011*, 2011. URL `http://ufldl.stanford.edu/housenumbers/nips2011_housenumbers.pdf`.

Tongzhou Wang, Jun-Yan Zhu, Antonio Torralba, and Alexei A Efros. Dataset distillation. *arXiv preprint arXiv:1811.10959*, 2018.

Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms, 2017.

Bo Zhao, Konda Reddy Mopuri, and Hakan Bilen. Dataset condensation with gradient matching, 2021.