Evaluating data reduction techniques for supervised training

Yuwen Heng

Master of Science
Data Science, Technology, and Innovation
School of Informatics
University of Edinburgh
2020

Abstract

Training deep neural networks can be resources-consuming. The budget required is increasing with the size of the dataset. During the past few decades, many research is dedicated to developing training procedures to accelerate the convergence speed of deep learning. However, we still need the whole dataset to train the network and paying for a large dataset may not pay back well if we can use a smaller subset to achieve an acceptable performance. To solve this issue, we first adapted and evaluated three methods, Patterns by Ordered Projections (POP), Enhanced Global Density-based Instance Selection (EGDIS), and Curriculum Learning (CL), to reduce the size of two image datasets, CIFAR10 and CIFAR100, for the classification task. Based on the analysis, we present our two contributions: the Weighted Curriculum Learning (WCL) and a trade-off framework. The WCL outperforms POP and EGDIS in terms of both classification accuracy and time complexity. It achieves comparable performance compared with CL while keeping a portion of hard examples. The trade-off framework selects a subset of samples according to the acceptable relative accuracy and the dataset. In addition, the framework is also extended to predict the number of samples needed to achieve a particular accuracy with a given subset.

Acknowledgements

First and foremost, I would like to express my deepest gratitude to my supervisor, Dr Yang Cao, for offering me the opportunity to work with him on such an attracting and challenging project. His encouragement and valuable guidance helped me tackle the obstacles in my research path.

Furthermore, special thanks go to Professor Bob Fisher, Dr Pavlos Andreadis at the University of Edinburgh and Dr Jiacheng Ni at IBM for sharing me their knowledge about computer vision and deep learning. The programming skills and coursework experience that I learnt from them helped me to organise the experiments well.

Finally, I would like to send my love to my fiancee Danni Li for her accompany during the past three years . I wouldn’t have the chance to study full-time without her full support.

Table of Contents

Chapter 1

Introduction

1.1 Motivation

1.2 Research Goal

1.3 Significance

1.4 Beneficiaries

Chapter 2

Background Research

In this chapter, we begin with presenting the necessary background to understand the supervised learning and data reduction methods, as well as, other ideas required to understand our research method. We we start with the structure of CNN and the training procedure. We then discuss the modern subset selection methods that can speed up the training procedure and outline their deficiencies. Next, we review the data reduction literature and present a CNN data reduction framework - use the network pre-trained on ImageNet to extract low-dimensional features and run the data reduction methods on extracted features. Furthermore, we cover the existing trade-off framework BlinkML [16] in the context of maximum-likelihood estimation machine learning algorithms and explain why it is not suitable for deep neural network. Finally, we present TAPAS [6], which is an accuracy predictor for deep neural network without training and has several properties that make it useful to build our trade-off framework.

2.1 Supervised Learning with CNN

CNN based supervised learning is a kind of machine learning task which learns the mapping between input visual data and output based on a set of well-labelled training samples. The visual data can be images, videos or even 3D models [19]. The output score after the softmax operation can be considered as the probability for a given image belongs to each class, $P(class|image, network)$. The CNN itself can be considered as a set of chained operations with trainable parameters. These parameters define the actual input-out mapping. For this reason, we use the symbol $f(x|\theta)$ to represent the output score predicted by the CNN which takes the input x with a particular parameter set θ. Figure 2.1 gives a basic CNN structure which is designed to classify images as cats or dogs. It contains two convolutional (Conv) layers, one max pooling layer and one fully connected (FC) layer. The FC layer is actually a multi-class logistic regression model which maps the outputs of the max pooling layer to the class scores. From this perspective, we can divide the CNN structure into two parts: feature extraction part and logistic regression part. The feature extraction part performs as a blackbox which transforms the input images to points in a lower-dimensional, linearly separable space.

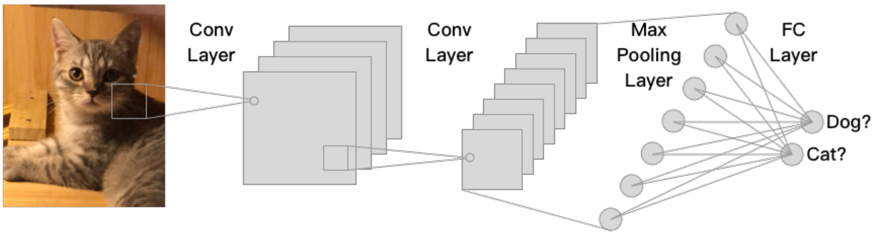

Figure 2.1: A basic CNN structure to classify images between cats and dogs. The outputs of the penultimate layer are extracted lower-dimensional features of the input images. These features should be linearly separable to achieve a high classification accuracy

If we use the symbol y to represent the ground truth of the input sample x, use $L(f(x|\theta),y)$ to represent the loss function which measures the difference between the predicted output and the ground truth label, then the training process is to find the parameter set θ^* which minimise the average loss of the whole training set as fellows:

$$\theta^* = \arg\min_{\theta} \frac{1}{N}\sum_{i=1}^{N} L(f(x_i|\theta),y_i) \tag{2.1}$$

where the symbol N stands for the number of samples in the training set. This turns the training process into an optimisation problem. Different from machine learning algorithms like logistic regression and support vector machine, the equation 2.1 is non-convex thus cannot be solved analytically [3, p. 304]. A number of techniques have been developed to solve the problem with the requirement that the loss function $L(.,.)$ is continuous. The basic one to train on large dataset is called stochastic gradient descent (SGD) which updates the parameters with the partial derivatives of a randomly selected sample. At each step, the new parameter is calculated with

$$\theta_{t+1} = \theta_t - \eta \frac{\partial L(f(x|\theta),y)}{\partial \theta_t} \tag{2.2}$$

and η is the step size whose typical value is between 0.1 to 0.001. A simple variant of SGD is mini-batch gradient descent which divides the training set into disjoint subsets and averages the gradients within the subset before updating the parameters:

$$\theta_{t+1} = \theta_t - \eta \frac{1}{M} \sum_{i=1}^{M} \frac{\partial L(f(x_i|\theta), y_i)}{\partial \theta_t} \tag{2.3}$$

where M is the batch size of the subset. For CNN, batch size M is often smaller than the training set size N because it takes too much memory to fit the whole dataset. Usually we use 128 or 256 as the batch size.

2.2 Mini-batch Sampling

Since the mini-batch gradient method trains the network with a subset of samples at each step, how to select the samples becomes a problem in the deep learning literature. Instead of uniform sampling, many researchers proposed to rank the samples with importance score or classification difficulty and selects a mini-batch based on different criteria. According to [4], we can divide the sampling methods into two categories: **current hypothesis method** and **targer hypothesis method**. Current hypothesis method measures the samples based on the parameter set θ_t at step t while targer hypothesis method is based on the final parameter set θ^*.

2.2.1 Current Hypothesis Method

Different authors have proposed a variety of current hypothesis methods. Specifically, in self-paced learning [11, 12, 15], active bias learning [2], and hard example mining [18, 13], the scores are calculated based on the sample difficulty, which is proportional to the classification score of the true class, $f(x|\theta, y)$. For importance sampling methods, the scores are calculated based on the gradient norm for each sample, $|\frac{\partial L(f(x_i|\theta), y_i)}{\partial x_i}|$. We finish this section by briefly explaining the actual implementations of these approaches.

2.2.1.1 Difficulty Based Method

Self-paced learning method tends to select easy samples which have a high classification score by injecting a pace function into the optimisation target function 2.1:

$$\theta^* = \arg\min_{\theta, v} \sum_{i=1}^{N} v_i L(f(x_i|\theta), y_i) + \lambda \sum_{i=1}^{N} v_i \tag{2.4}$$

where v is the score calculated by the pace function. The pace function can be either a simple step function [11] or a more complicated dynamic function which changes with step t [12] as long as it can assign value 0 to samples. By minimising the target function 2.4, the method would zero out hard examples which have higher loss L thus keep only the easy samples. This makes the trained network more robust to outliers [15].

A potential problem of self-paced learning is that it would gradually increase the loss of hard examples [2]. The possible solution is to use the active bias learning method, which is designed to select the uncertain samples whose classification score vary near the decision threshold. Chang et al. proposed and evaluated many self-paced methods and the representative one is called SGD Sampled by Threshold Closeness (SGD-STC) [2]. It records the historical average classification probability $\bar{P}$ for each sample and the score is calculated with a equation that is proportional to $(1-\bar{P}) \times \bar{P}$. However, the problem is that we need extra space and computation to maintain the historical scores.

Hard example mining is yet another heuristic method aims at maximising the convergence speed by extending the self-paced learning method [18]. The algorithm proposed by [13] ranks the samples based on the latest computed classification score in descending order. At early training stages, the algorithm chooses easy samples just like self-paced learning. After a thorough exploitation process, the algorithm tends to select hard examples which have low classification scores.

2.2.1.2 Importance Based Method

Although published experiments in the cited resources above prove that difficulty based methods can surely speed up the training process and may achieve even higher accuracy, the lack of mathematical prove could lower the interests of researchers. In the contrary, importance based method raises from the profound mathematical demonstration [20] and is more reliable. Despite the elaborate derivation, the most important conclusion is that the optimal weight distribution is proportional to the per sample gradient norm.

The challenge is that computing the per sample gradient norm $|\frac{\partial L(f(x_i|\theta),y_i)}{\partial x_i}|$ is intractable. In the past few years, many researchers have adapted their approximate methods to speed up the process. The most convincing one is proposed by Katharopoulos et al. which derives an upper bound of the gradient norm [8],

$$|\frac{\partial L(f(x_i|\theta),y_i)}{\partial x_i}| \leq |h(x_i)| \tag{2.5}$$

that $h(x_i)$ is the upper bound function depends on the last layer pre-activation outputs and time step t. With this equation, we can compute the largest sample gradient after a single forward propagation.

The benefits of current hypothesis methods is that the sample importance varies with time step thus the chosen samples at each step can reflect the current capacity of the network. However, because evaluating the whole training set is time-consuming, we often select a subset uniformly first and then select the samples within the subset. This would affect the optimal theory performance.

2.2.2 Target Hypothesis Method

Compared with current hypothesis method, target hypothesis method selects instances based on the possible final performance of the network thus the weights of the samples are pre-defined and won't change during the training process [1]. For this reason, target hypothesis methods are more suitable to reduce the size of the dataset. To our knowledge, Curriculum Learning (CL) is the only method with these properties.

Similar with hard example mining, CL trains the network with easy samples first then adds more difficult samples into the dataset and gradually the subset would contain all the training samples. The main difference is that the difficulties of the samples are measured in advance, whether with a pre-trained network or with a linear classifier like SVM [4].

2.3 Data Reduction Algorithm

Data reduction algorithm becomes famous with the increasing of dataset size and compute time. The most common one is called Principal Components Analysis (PCA) which projects the features onto the most important few eigenvectors to capture the most variants. The famous example eigenface proves that PCA is efficient even with image dataset [9]. Apart from PCA, there are other methods such as dimensionality reduction methods which reduce the size of the sample features and instance selection methods that can reduce the number of samples in the training set. We discuss some typical implementations in the next subsections.

2.3.1 Dimensionality Reduction

As discussed above, PCA is the most common method for both structured dataset and unstructured dataset. In the context of deep CNN, however, the term feature extraction is more famous as discussed in section 2.1. Kornblith et al. evaluated the extracted features with pre-trained networks on many vision datasets and the results show that the logistic regression accuracy is linearly related to the ImageNet classification accuracy [10]. The problem is that

2.3.2 Instance Selection

2.4 Trade-off Framework

2.5 Accuracy Predictor

Chapter 3

Adapted Data Reduction Methods

In this chapter, we begin by presenting the pre-processing feature extraction process for image dataset. Next we adapt three methods overviewed in Chapter 2 to reduce the size of image dataset, called the Patterns by Ordered Projections (POP) [17], Enhanced Global Density-based Instance Selection (EGDIS) [14], and Curriculum Learning (CL) [4]. Then we propose our weighted data reduction method, called Weighted Curriculum Learning (WCL), based on CL scores and the EGDIS selected boundary instances. We also illustrate the selection patterns with three generated blob datasets, which correspond to the 2-dimensional special case of extracted image feature space. After that, our work is focused on the comprehensive evaluation of the methods. We describe image augmentation algorithms and the details of the DenseNet architecture [5] and incremental training [7]. We also describe the model fitting procedure of the SVM-baseline.

3.1 Image Feature Extraction

3.2 Patterns by Ordered Projections

3.3 Enhanced Global Density-based Instance Selection

3.4 Curriculum Learning

3.5 Weighted Curriculum Learning

3.6 Evaluation Designs

Chapter 4

Data Reduction Evaluations

4.1 Time Complexity

4.2 Classification Accuracy

Chapter 5

Trade-off Framework

5.1 Subset Selection Framework

Chapter 6

Trade-off Evaluation

6.1 Relative Accuracy Precision

Chapter 7

Conclusion and Future Work

Bibliography

[1] Yoshua Bengio, Jérome Louradour, Ronan Collobert, and Jason Weston. Curriculum learning. In *ACM International Conference Proceeding Series*, volume 382, pages 1–8, New York, New York, USA, 2009. ACM Press.

[2] Haw Shiuan Chang, Erik Learned-Miller, and Andrew McCallum. Active bias: Training more accurate neural networks by emphasizing high variance samples. In *Advances in Neural Information Processing Systems*, volume 2017-Decem, pages 1003–1013, 2017.

[3] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. 2016.

[4] Guy Hacohen and Daphna Weinshall. On the power of curriculum learning in training deep networks. In *36th International Conference on Machine Learning, ICML 2019*, volume 2019-June, pages 4483–4496, 2019.

[5] Gao Huang, Zhuang Liu, Laurens van der Maaten, and Kilian Q. Weinberger. Densely Connected Convolutional Networks. In *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*, volume 2017-Janua, pages 2261–2269. Institute of Electrical and Electronics Engineers Inc., aug 2016.

[6] R. Istrate, F. Scheidegger, G. Mariani, D. Nikolopoulos, C. Bekas, and A. C. I. Malossi. TAPAS: Train-Less Accuracy Predictor for Architecture Search. *Proceedings of the AAAI Conference on Artificial Intelligence*, 33:3927–3934, jun 2019.

[7] Roxana Istrate, A. C.I. Malossi, Costas Bekas, and Dimitrios Nikolopoulos. Incremental training of deep convolutional neural networks. In *CEUR Workshop Proceedings*, volume 1998. CEUR-WS, mar 2017.

[8] Angelos Katharopoulos and François Fleuret. Not All Samples Are Created Equal: Deep Learning with Importance Sampling. *35th International Conference on Machine Learning, ICML 2018*, 6:3936–3949, mar 2018.

[9] Ramandeep Kaur and Er Himanshi. Face recognition using Principal Component Analysis. In *Souvenir of the 2015 IEEE International Advance Computing Conference, IACC 2015*, pages 585–589, 2015.

[10] Simon Kornblith, Jonathon Shlens, and Quoc V. Le. Do better imagenet models transfer better? In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 2019-June, pages 2656–2666, may 2019.

[11] M. Pawan Kumar, Benjamin Packer, and Daphne Koller. Self-paced learning for latent variable models. In *Advances in Neural Information Processing Systems 23: 24th Annual Conference on Neural Information Processing Systems 2010, NIPS 2010*, pages 1189–1197, 2010.

[12] Hao Li and Maoguo Gong. Self-paced Convolutional Neural Networks. Technical report, 2017.

[13] Ilya Loshchilov and Frank Hutter. Online Batch Selection for Faster Training of Neural Networks. nov 2015.

[14] Mohamed Malhat, Mohamed El Menshawy, Hamdy Mousa, and Ashraf El Sisi. A new approach for instance selection: Algorithms, evaluation, and comparisons. *Expert Systems with Applications*, 149:113297, jul 2020.

[15] Deyu Meng, Qian Zhao, and Lu Jiang. What Objective Does Self-paced Learning Indeed Optimize? 2015.

[16] Yongjoo Park, Jingyi Qing, Xiaoyang Shen, and Barzan Mozafari. BlinkML: Efficient maximum likelihood estimation with probabilistic guarantees. In *Proceedings of the ACM SIGMOD International Conference on Management of Data*, pages 1135–1152, New York, New York, USA, jun 2019. Association for Computing Machinery.

[17] José C. Riquelme, Jesús S. Aguilar-Ruiz, and Miguel Toro. Finding representative patterns with ordered projections. *Pattern Recognition*, 36(4):1009–1018, apr 2003.

[18] Abhinav Shrivastava, Abhinav Gupta, and Ross Girshick. Training region-based object detectors with online hard example mining. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 2016-Decem, pages 761–769, 2016.

[19] Wei Song, Lingfeng Zhang, Yifei Tian, Simon Fong, Jinming Liu, and Amanda Gozho. CNN-based 3D object classification using Hough space of LiDAR point clouds. *Human-centric Computing and Information Sciences*, 10(1):1–14, dec 2020.

[20] Peilin Zhao and Tong Zhang. Stochastic optimization with importance sampling for regularized loss minimization. In *32nd International Conference on Machine Learning, ICML 2015*, volume 1, pages 1–9, 2015.