GPU-based Training of Autoencoders for Bird Sound Data Processing

Jian Guo*, Kun Qian†, Björn Schuller‡ and Satoshi Matsuoka*
*Matsuoka Lab, Tokyo Institute of Technology, Japan, Email: guo.j.ae@m.titech.ac.jp
†MISP group, MMK, Technische Universität München, Germany, Email: andykun.qian@tum.de
‡Department of Computing, Imperial College London, UK, Email: bjoern.schuller@imperial.ac.uk

***Abstract*—Bird sounds have been studied in recent years due to their significance in helping ornithologists, and ecologists to monitor birds activities, which reflect climate changes, biodiversity, and reserves local protection status. Within the increasingly collected large amount of bird sound data from experts and amateurs, how to handle, and employ the state-of-the-art deep learning methods to mining such large amount of data, is bringing a huge challenge, and opportunity for the research community. In this work, we propose a framework using the GPU to accelerate autoencoders training for a large amount of bird sound data. Experimental results show that the GPU can considerably speed up the training process of bird sounds when fed within different scales of data, or feature numbers, compared with CPU-based learning.**

I. Introduction

Bird sounds, regarded as the 'speech of birds', are significant not only for ornithologists in studying birds' species, mate, activities, etc. [1], but also for the measurement of climate changes, and biodiversity of a reserve by ecologists [2]. With the increasingly collected amount of bird sound data by experts and amateurs, researchers in the recently highly active areas of 'deep learning', can contribute by digging such large amount of bird sound data efficiently. Among the structures of deep networks, 'autoencoders' are essential elements in designing stacked autoencoders deep networks [3], which are also useful for unsupervised learning [4]. However, training autoencoders on large amounts of data is a time-consuming task for traditional CPU-based computing, which makes it difficult to handle the large amount of bird sound data. In this work, we start exploring the potential in performance employing the latest GPUs in training autoencoders to the named end. We compare the computing performance of a single CPU, and single GPU based system, respectively, within different scales of data numbers, and acoustic feature numbers derived from the bird sounds.

II. Database and Framework

As Fig. 1 shows, the audio recordings of the bird sounds are processed by signal processing techniques to extract acoustic features, which represent the salient characteristics of the birds' sounds. These features act as the 'input' of autoencoders for the subsequent training in our consideration. The autoencoders and their 'output' can be further used as supervised, unsupervised, or semi-supervised learning elements, which contribute to the bird sounds study.

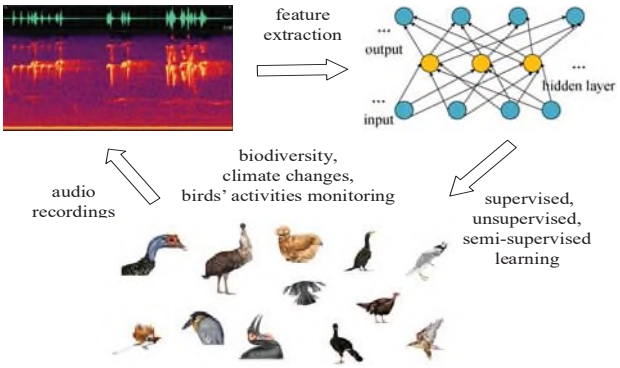


Fig. 1. *Training Autoencoders from a Large Amount of Bird Sound Data.*

A. 'LifeCLEF Bird task'

Since 2014, 'LifeCLEF'[1] [5] launched its first bird sounds classification task, which focused on bird identification by training 9 688 audio recordings from over 501 species from South America. Until recently in 2015, and 2016, 24 607 audio recordings (2015 and 2016 share the same training data) were used in the training phase for the recognition of 999 species of birds. Here, we use the training sets from 'LifeCLEF 2014' and 'LifeCLEF 2015' as Table I shows.

TABLE I
Bird Sound Data.

	species	recordings	size
'LifeCLEF 2014'	501	9 688	23.0 GB
'LifeCLEF 2015'	999	24 607	70.2 GB

TABLE II
openSMILE Feature Sets.

	dimensions
'eGeMAPS'	88
'IS09_emotion'	384
'IS13_ComParE'	6 373

B. Using openSMILE to Extract Audio Features

Here we use our toolkit, openSMILE [6], to extract a large scale of audio features from bird sound data, which was proved to be efficient in bird sound recognition [7], [8]. Features base on Low Level Descriptors (LLDs) with

[1]http://www.imageclef.org/2014/lifeclef/bird.

functionals form the input for training the autoencoders. We select three different kinds of feature sets from openSMILE, namely 'eGeMAPS', 'IS09_emotion' and 'IS13_ComParE' (refer to [6], [9]); the feature number per audio instance are given in Table II. These features were designed by audio experts, including pitch, formants, Mel-Frequency Cepstral Coefficients (MFCCs), spectral parameters, energy/amplitude related parameters, to name but a few. The sets represent different feature dimensions scaling at 10, 10^2, and 10^3, respectively.

III. Experiments and Discussion

We set up our experiments on training a stacked autoencoders' network [3] (256-256-256-501/999) with a learning rate of 0.1, and a batch size of 256. Autoencoders are feedforward neural networks, which aim at minimizing the reconstruction errors between the inputs $\boldsymbol{X}$, to the outputs $\widetilde{\boldsymbol{X}}$ as:

$$\boldsymbol{J}(\boldsymbol{X}, \widetilde{\boldsymbol{X}}) = ||\boldsymbol{X} - \widetilde{\boldsymbol{X}}||^{\mathbf{2}}, \tag{1}$$

where $\boldsymbol{X}$, $\widetilde{\boldsymbol{X}}$ represents the inputs, and outputs of the autoencoders respectively. In a simplest case, when feeding $\boldsymbol{X}$ into a one-hidden-layer autoencoder, a new map will be generated as:

$$\boldsymbol{Y} = \boldsymbol{\sigma_1}(\boldsymbol{W}\boldsymbol{X} + \boldsymbol{b}), \tag{2}$$

where $\boldsymbol{W}$ is the weight vector, and $\boldsymbol{b}$ is the bias. $\boldsymbol{\sigma}$ is the activation function. $\widetilde{\boldsymbol{X}}$ is reconstructed as:

$$\widetilde{\boldsymbol{X}} = \boldsymbol{\sigma_2}(\widetilde{\boldsymbol{W}}\boldsymbol{Y} + \widetilde{\boldsymbol{b}}). \tag{3}$$

Thus, 'autoencoders' can learn some higher level features from the original inputs, i. e. lower level features.

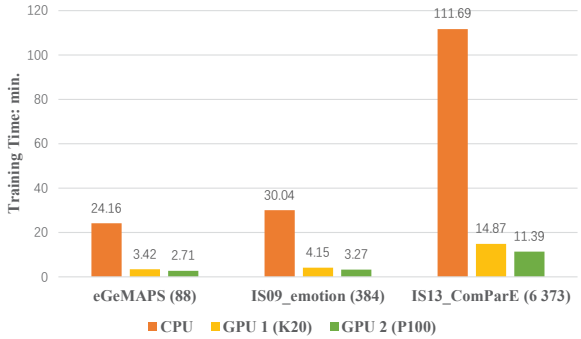
TABLE III
Hardware Configuration

	Hardware Configuration
CPU	Intel Core i7-6700K @4.0 GHz
GPU 1	Tesla K20Xm 6 GB GDDR5 with CUDA 8.0
GPU 2	Tesla P100 PCIE 16 GB HBM2 with CUDA 8.0

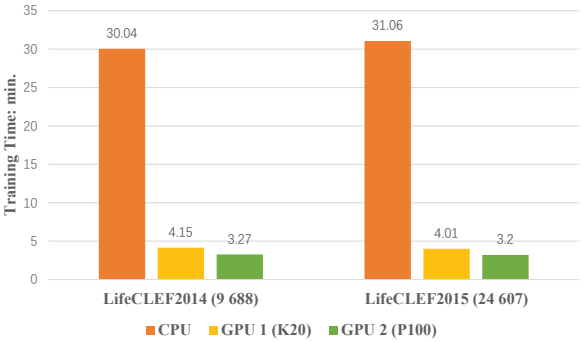
We conduct our experiments with hardware configurations shown as Table III. Results are shown in Fig. 2. We can see that, the GPU 1 (a K20) can achieve around 7-8 times and the newer GPU 2 (a P100) can achieve even almost 10 times the speedup vs the CPU Intel Core i7-6700K (4 Cores, 8 threads) when training with smaller feature numbers. In our experiments, the number of features has a more important impact on the time costs in training autoenocoers, rather than the number of data needed to be processed. It appears reasonable that, within the more complicated data features, the GPU will be one's first choice in future autoencoder training.

IV. Conclusion

In this work, we quantitatively compared the performance of CPU, and GPU on training autoencoders for a large amount of bird sound data. Our experimental results show that the GPU outperformed the CPU considerably, specifically, when the feature dimension, or data number is increased. Future work includes implementing deep learning solutions for big bird sound data processing on multiple GPUs, and multiple nodes of GPUs.



(a) Autoencoders Training with Different Feature Sets



(b) Autoencoders Training with Different Data Sets

Fig. 2. Performance of CPU and GPU Training of Autoencoders.

Acknowledgment

This work was partially supported by JST CREST Grant Number JPMJCR1303 and JPMJCR1687, Japan, and the China Scholarship Council (CSC), China.

References

[1] C. K. Catchpole and P. J. Slater, *Bird song: biological themes and variations*. Cambridge, UK: Cambridge university press, 2003.

[2] C. Parmesan and G. Yohe, "A globally coherent fingerprint of climate change impacts across natural systems," *Nature*, vol. 421, no. 6918, pp. 37–42, 2003.

[3] J. Gehring, Y. Miao, F. Metze, and A. Waibel, "Extracting deep bottleneck features using stacked auto-encoders," in *Proc. of IEEE ICASSP*. IEEE, 2013, pp. 3377–3381.

[4] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[5] H. Goëau, H. Glotin, W.-P. Vellinga, R. Planqué, A. Rauber, and A. Joly, "Lifeclef bird identification task 2014," in *Proc. of CLEF2014*, Sheffield, UK, 2014, pp. 585–597.

[6] F. Eyben, F. Weninger, F. Gross, and B. Schuller, "Recent developments in opensmile, the munich open-source multimedia feature extractor," in *Proc. of ACM MM*. Barcelona, Spain: ACM, 2013, pp. 835–838.

[7] K. Qian, Z. Zhang, F. Ringeval, and B. Schuller, "Bird sounds classification by large scale acoustic features and extreme learning machine," in *Proc. of GlobalSIP*. Orlando, Florida, USA: IEEE, 2015, pp. 1317–1321.

[8] K. Qian, Z. Zhang, A. Baird, and B. Schuller, "Active learning for bird sounds classification," *Acta Acustica united with Acustica*, vol. 103, pp. 361–364, 2017.

[9] F. Eyben, *Real-time speech and music classification by large audio feature space extraction*. Springer, 2015.