Unsupervised Deep Learning with Stacked Autoencoders on Chameleon

Pablo Rivas[1]*, Ezequiel Rivas[2], Deep Dand[1], Raul Aragon[3]

Chameleon project: CH-818931, “Research on applications of sparse stacked deep autoencoders”
Project type: research
Resource type: bare metal/CHI@TACC

[1] *Department of Computer Science, Marist College, New York, USA*
[2] *Department of Graduate Studies and Research, Nogales Institute of Technology, Nogales, Mexico*
[3] *Computing Center, Nogales Institute of Technology, Nogales, Mexico*
***Main contact**: Pablo.Rivas@Marist.edu

Abstract

Description of Research Problems

We are investigating the applications of deep autoencoders in computer vision and image processing. Specifically we have been studying models with more than five layers of unsupervised encoder-decoder and a softmax layer in the output. We aim to show that simple unsupervised deep nerual models such as this can perform as good as other deep learning strategies while being more efficient than other GPU-based learning algorithms. This research, in the bigger picture, addresses the misconception that convolutional-based neural network architectures are the panacea in the computer vision and image processing in our days.

There is a particular problem we are addressing to prove our hypothesis; we address the problem of hand gestures recognition in the American Sign Language (ASL) using unsupervised deep learning. We found that even a simple five layer model using unsupervised encoder-decoder neural units shows outstanding performance. We use a dataset of segmented images captured with a depth-sensor camera for different subjects [1]. The average accuracy obtained was of 98.9% and as high as 99.4% on unseen data.

Description of Experiments

Our experiments consist of writing Python code to use the TensorFlow (TF) API made available for this programming language. The primary reason for its usage was the accessibility to use the CUDA-enabled libraries in an abstract level. Thus, TF is a tool to use the GPUs in our Chameleon appliance. We selected an appliance loaded with a pre-built image with CUDA 8.0 libraries installed on CentOS 7.

This appliance is based on the default CentOS 7 which has been customized to provide the NVIDIA drivers and the CUDA toolkit. CUDA version 8+ is required for the NVIDIA P100 GPUs. The image name is `CC-CentOS7-CUDA8`. According to the documentation, this appliance is built from the GPU branch of `CC-CentOS7` disk image builder.

This CentOS image was sufficient for out first experiments, and we personalized it to have specific python packages and libraries necessary to run CUDA-intensive learning algorithms in TF. However, after our initial experiments, it was made available a new image running Ubuntu 16.04 with CUDA 8.0, and knowing the larger support for Ubuntu systems with CUDA libraries we decided to continue our experiments on this new image. Soon we realized that the Ubuntu-based appliance required more setup to have TF and CUDA operating smoothly over Python, and since our experiments would run for 10-14 days it was necessary to create a step-by-step setup guide to install all the necessary packages, libraries, configurations, and permissions to specific CUDA libraries and files. This setup script has been made available in our GitHub repository, and it applies only to the image used, which is named `CC-Ubuntu16.04-CUDA8`.

The most important things that we installed for the customization of our leased host and to run our experiments were the following:

Libraries

- `cuDNN`: These are libraries from NVIDIA. They are known as CUDA Deep Neural Network (cuDNN) libraries and are a GPU-accelerated set of primitives for deep neural networks.
- `libatlas-dev`: Pre-compiled libraries for linear algebra operations.

Packages

- `gcc gfortran`: C/C++ compiler for compiling libraries locally.
- `python-{numpy scipy matplotlib}`: Python libraries for scientific computing, mathematical processing of *n*-dimensional arrays, and production of plots.
- `tensorflow`: This is Google’s main TF GPU version for linux 64-bit systems.
- `glances, nvidia-ml-py, screen`: For locally and remotely monitoring the system resources, including NVIDIA GPUs, and to allow multiple sessions to run upon disconnection.

Leases

We leased hosts under the category of `GPU P100` systems. This gives us access to two GPU units that we use for the intensive *unsupervised* training of deep neural networks.

Acknowledgments

Any results presented here were in part obtained using the Chameleon testbed supported by the National Science Foundation.

References

[1] Byeongkeun Kang, Subarna Tripathi, and Truong Q Nguyen. Real-time sign language fingerspelling recognition using convolutional neural networks from depth map. In *Pattern Recognition (ACPR), 2015 3rd IAPR Asian Conference on*, pages 136–140. IEEE, 2015.