Chainer-XP: A Flexible Framework of Neural Networks for the Intel® Xeon Phi™ Coprocessor

T. D. Diep[1], M. T. Nguyen[1], N.-Y Nguyen-Huynh[1], M. T. Chung[1], M. T. Nguyen[1], N. Q. Hung[1], and N. Thoai[1]

Abstract: Artificial Neural Networks (ANNs), which is inspired by the biological neural networks, has attracted a lot of community attention by learning to do tasks by considering examples rather than task-specific programming. Such networks are applied to a wide range of research fields like computer vision, speech recognition, quantum chemistry, social network filtering, medical diagnosis. Hence, there are several frameworks have been created, which helps scientists and researchers set up ANNs straightforwardly, such as Caffe, Theano, TensorFlow, Chainer. In all aforementioned frameworks, Chainer is the only one defined on-the-fly via the actual forward computation compared with the others. Thus, Chainer enables its users to fully leverage the power of programming logic in Python.

Chainer is a well-known framework for ANNs enabling users to write complex architectures simply and intuitively. However, Chainer in particular as well as others in general are used effectively on only Central Processing Units (CPUs) along with Graphics Processing Units (GPUs). On the other hand, there are many contemporary systems consisting of Intel Xeon processors and Intel Xeon Phi coprocessors without GPUs but almost existing frameworks for ANNs cannot be run on such infrastructures excluding Knights Landing. As a result, there is a considerable demand for developing one framework which can be run on such systems. Nevertheless, there are some enormous challenges that we will face when developing one such framework. One typical example is that the user interface of Chainer uses Python programming language which the coprocessors do not support. Fortunately, there is a convenient module called pyMIC that assists programmers to be able to handle offloads to the coprocessors from Python code. Furthermore, pyMIC also provides some useful functions to handle data transfers in a flexible yet performant way.

Our promising approach makes use of pyMIC to integrate it into Chainer in order to create a novel framework named Chainer-XP which can be run effectively on systems including CPUs connected with the coprocessors. We have tried to run Chainer-XP using MNIST popular handwritten digit database from scratch. Preliminary results show that Chainer-XP is capable of being operated on the system comprising two Intel Xeon processors and one Intel Xeon Phi coprocessor. Nonetheless, Chainer-XP still poses some problems about the performance aspect. Further, It is highly probable that pyMIC has several latent bugs due to the profound lack of the maintenance. Therefore, we intend to not only investigate but also optimize both Chainer-XP and pyMIC so as to improve the performance dramatically.

[1] HPC Lab, HCMC University of Technology, 268 Ly Thuong Kiet, Ho Chi Minh City, Vietnam
{*minhtri, hungnq2*}*@cse.hcmut.edu.vn*
{*dang, nhny, ctminh, nmthin, namthoai*}*@hcmut.edu.vn*