September 2017

Implementing Deep Neural Networks on Fresh Breeze

JACK B. DENNIS [a], LEI HUANG [b,1], WILLIE LIM [a], HSIANG-HUANG WU [b], and YUZHONG YAN [b],

[a] *MIT CSAIL*

[b] *Prairie View A & M University*

Abstract. This paper discusses how deep neural network simulations can be implemented and executed on a proposed computer system inspired by data flow principles. In particular, we use the *Kiva* simulator developed by the MIT CSAIL Fresh Breeze project to simulate the data flow machine used for running the neural networks. An example of a deep neural network and its simulation using Kiva is presented. The neural network is written in *funJava*, a functional version of Java. Our compiler for funJava programs converts funJava methods to data flow graphs, identifies opportunities for data parallel implementation, and generates a graph of machine executable code blocks called *codelets* which are executed as tasks by the data flow machine. The behavior and performance of the neural network as the input data size and the number of cores in the data flow machine are varied are reported and discussed.

Keywords. Deep Learning, Neural Network, Data Flow Machine

1. Introduction

Big data analytics research has become one of the hottest topics in both industry and academia due to its ability to turn data into knowledge and revenue. With advances in machine learning [14], especially in deep learning, big data analytics platforms are now able to automatically extract more valuable insights from the data [12,11]. These platforms need to meet the high demands of data storage and computing power to handle ever-growing volumes of data and the expensive deep learning training process. Streaming data analytics are widely used for mission critical applications where timely completion of data analytics tasks is crucial. We need to optimize big data analytics platforms, especially for the deep learning training process which takes most of the time. In this paper, we discuss a new deep neural network implementation inspired by the data flow execution model and present simulation results showing its performance and scalability.

For our implementation, we express the machine learning application in funJava, a functional subset of Java. funJava programs are compiled and then loaded into and executed by a simulated Fresh Breeze multi-core processor [6,7] with features designed to realize high performance fine-grain multi-tasking. All funJava vectors and arrays are

[1] Corresponding Author: Prairie View A&M University, SR Collins Room 314, MailStop: 2515, Prairie View, TX 77446, USA; E-mail: lhuang@pvamu.edu