Available online at www.sciencedirect.com
ScienceDirect
Computer Speech & Language xxx (2017) xxx-xxx
www.elsevier.com/locate/csl

Acoustic model training based on node-wise weight boundary model for fast and small-footprint deep neural networks☆,☆☆

Q1 Ryu Takeda*,[a], Kazuhiro Nakadai[b], Kazunori Komatani[a]

[a] The Institute of Scientific and Industrial Research, Osaka University, 8-1, Mihogaoka, Ibaraki, Osaka 567-0047, Japan
[b] Honda Research Institute Japan Co., Ltd., 8-1, Honcho, Wako, Saitama 351-0114, Japan

Received 8 April 2016; received in revised form 31 January 2017; accepted 3 February 2017

Abstract

Our goal for this study is to enable the development of discrete deep neural networks (NNs), some parameters of which are discretized, as small-footprint and fast NNs for acoustic models. Three essential requirements should be met for achieving this goal; 1) the reduction in discretization errors, 2) implementation for fast processing and 3) node-size reduction of DNNs. We propose a weight-parameter model and its training algorithm for 1), an implementation scheme using a look-up table on general-purpose CPUs for 2), and a layer-biased node-pruning method for 3). The first proposed method can set proper boundaries of discretization at each NN node, resulting in reduction in discretization errors. The second method can reduce the memory usage of NNs within the cache size of the CPU by encoding the parameters of NNs. The last method can reduce the network size of the quantized DNNs by measuring the activity of each node at each layer and pruning them with a layer-dependent score. Experiments with 2-bit discrete NNs showed that our training algorithm maintained almost the same word accuracy as with 8-bit discrete NNs. We achieved a 95% reduction of memory usage and a 74% increase in speed of an NN's forward calculation.

Keywords: Deep neural network; Acoustic model; Small-footprint; Quantization; Node-pruning

1. Introduction

1.1. Background

Deep neural networks (DNNs) are widely used as acoustic models in automatic speech recognition (ASR) instead of Gaussian mixture models (GMMs) because of their high word accuracy (WA) (Dahl et al., 2012; Hinton et al., 2012; Seide et al., 2011a; 2011b). However, the applicable computer architecture of DNNs is still restricted because of their high computational cost. Although implementation using a graphics processing unit (GPU) (Raina et al., 2009) and distributed computing (Dean et al., 2012) is effective for increasing the speed of DNNs, such an expensive

☆ This paper is a modified and extended version of our earlier reports (Takeda et al., 2015)
☆☆ This paper has been recommended for acceptance by Prof. R. K. Moore.
* Corresponding author.
E-mail address: rtakeda@sanken.osaka-u.ac.jp (R. Takeda).

http://dx.doi.org/10.1016/j.csl.2017.02.002

approach cannot be applied to resource-restricted systems, such as small/micro computers or embedded systems with ordinary CPUs. Thus, small-footprint DNNs in terms of memory usage and computational cost are required.

Parameter discretization can drastically reduce memory usage and computational cost; thus, enabling DNNs to finish processing within a reasonable time without GPUs and distributed computing. A fixed-point DNN, whose weights, bias parameters, and middle layer inputs are linearly quantized to n bits, enables fast processing on a very large-scale integration (Kim et al., 2014) or a CPU with the Supplemental Streaming SIMD Extensions 3 (SSSE3) instruction set (Vanhouche et al., 2011). The parameters of fixed-point DNNs are trained by iterating the quantization of weights to n bits and usual back propagation (Kim et al., 2014; Tang and Kwang, 1993). However, experimental selection of n for best performance requires a massive number of experiments. Moreover, the above implementation still requires a special instruction set of CPUs or special devices. Neural networks implemented on general-purpose CPUs without a special instruction set will be helpful for small/micro computers. We can also develop embedded systems using a field-programmable gate array with a CPU[1].

The use of look-up tables (LUTs) is a promising approach, and we call NNs 'discrete NNs' because their weights are treated as an encoded address for an LUT. For example, the speed of NNs can increase by SSSE3-like processing with an LUT on a general-purpose CPU by exploiting its cache memory. Of course, such techniques can also be applied to hardware. Our previously proposed training algorithm is based on a weight boundary model and control of the weight boundary (Takeda et al., 2014). With this model, the weight boundary at each layer is restricted to a certain range (*layer-wise* weight boundary model), and plays the role of layer-wise weight normalization. Weights are discretized only once after training because the distribution of weights has already become uniform or Bernoulli through training. We confirmed that 4-bit discrete NNs actually worked without degradation in a large vocabulary ASR task. A critical issue with 4-bit discrete NNs is that the use of an LUT for achieving fast processing is not realistic because the table size becomes more than 32 Mbytes, at which a CPU cache no longer works well. Therefore, smaller-bit, such as 2 bits, discrete NNs are required.

We propose a parameter training algorithm and implementation scheme based on our newly developed *node-wise* weight boundary model to achieve 2-bit discrete NNs. The key for 2-bit discretization is the fact that our previous model normalized weights at each layer in quantization. The quantization errors with layer-wise normalization increase because the dynamic range and distribution of weights differ at each node. Therefore, node-wise normalization is expected to adjust the weight's range not for a set of all nodes in the layer but for each individual node. Such normalization is incorporated in our node-wise weight boundary model. We also developed two weight encoding methods using LUTs, one of which can dramatically reduce the LUT size.

We also propose a *layer-biased* node-pruning method of DNNs to compress the size of DNNs after using our quantization method. This pruning is applied only once after our proposed quantization, and the number of weight parameters is reduced dramatically. Since our pruning method prunes the nodes of the last layer that has many weight parameters prior to other layers, it also contributes to increasing the speed of forward calculation unlike the *layer-flat* pruning method (He et al., 2014). This combination of our proposed pruning and quantization methods reduces memory usage and increases processing speed efficiently. Our training algorithm, LUT-implementation, and node-pruning method were validated through experiments involving a large vocabulary ASR task with clean speech signals and a real-time factor (RTF) of forward calculation of NNs. The WAs of reverberant and noisy speech signals were also investigated for performance analysis under a cross-environment and a mismatched condition of acoustic model.

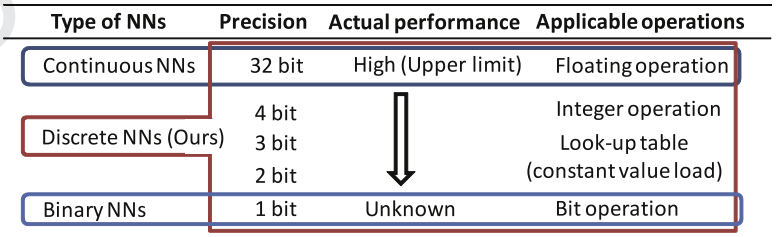

Fig. 1. Properties of neural networks.

[1] such as http://www.xilinx.com/products/silicon-devices/soc/zynq-ultrascale-mpsoc.html

Please cite this article as: R. Takeda et al., Acoustic model training based on node-wise weight boundary model for fast and small-footprint deep neural networks, Computer Speech & Language (2017), http://dx.doi.org/10.1016/j.csl.2017.02.002