DQRM: Deep Quantized Recommendation Model A Recommendation Framework That is Small, Powerful, and Efficient to Train

Ben Trovato*
G.K.M. Tobin*
trovato@corporation.com
webmaster@marysville-ohio.com
Institute for Clarity in Documentation
Dublin, Ohio, USA

Lars Thørväld
The Thørväld Group
Hekla, Iceland
larst@affiliation.org

Valerie Béranger
Inria Paris-Rocquencourt
Rocquencourt, France

Aparna Patel
Rajiv Gandhi University
Doimukh, Arunachal Pradesh, India

Huifen Chan
Tsinghua University
Haidian Qu, Beijing Shi, China

Charles Palmer
Palmer Research Laboratories
San Antonio, Texas, USA
cpalmer@prl.com

John Smith
The Thørväld Group
Hekla, Iceland
jsmith@affiliation.org

Julius P. Kumquat
The Kumquat Consortium
New York, USA
jpkumquat@consortium.net

ABSTRACT

A clear and well-documented LaTeX document is presented as an article formatted for publication by ACM in a conference proceedings or journal publication. Based on the "acmart" document class, this article presents and explains many of the common variations, as well as many of the formatting elements an author may use in the preparation of the documentation of their work.

CCS CONCEPTS

• **Computer systems organization → Embedded systems**; *Redundancy*; Robotics; • **Networks** → Network reliability.

KEYWORDS

datasets, neural networks, gaze detection, text tagging

ACM Reference Format:
Ben Trovato, G.K.M. Tobin, Lars Thørväld, Valerie Béranger, Aparna Patel, Huifen Chan, Charles Palmer, John Smith, and Julius P. Kumquat. 2018. DQRM: Deep Quantized Recommendation Model A Recommendation Framework That is Small, Powerful, and Efficient to Train. In *Proceedings of Make sure to enter the correct conference title from your rights confirmation emai (Conference acronym 'XX).* ACM, New York, NY, USA, 4 pages. https://doi.org/XXXXXXX.XXXXXXX

1 INTRODUCTION

ACM's consolidated article template, introduced in 2017, provides a consistent LaTeX style for use across ACM publications, and incorporates accessibility and metadata-extraction functionality necessary for future Digital Library endeavors. Numerous ACM and SIG-specific LaTeX templates have been examined, and their unique features incorporated into this single new template.

If you are new to publishing with ACM, this document is a valuable guide to the process of preparing your work for publication. If you have published with ACM before, this document provides insight and instruction into more recent changes to the article template.

The "`acmart`" document class can be used to prepare articles for any ACM publication — conference or journal, and for any stage of publication, from review to final "camera-ready" copy, to the author's own version, with *very* few changes to the source.

2 EXPERIMENT RESULTS

In this section, we present our results evaluating our DQRM framework on two commonly used click through rate recommendation datasets and two DLRM model configurations accordingly. The two datasets are Criteo Kaggle Display Advertising Challenge Dataset (shortened below as Kaggle dataset) and Criteo Terabyte Dataset (shorted as Terabyte dataset). The original DLRM work [3] proposes two different configurations for each dataset, and we followed these two configurations in our implementation of DQRM. We organize the experiments results under three sections, x.1, x.2, and x.3. Section x.1 details experiments on embedding table quantization alone evaluated on the Kaggle dataset. Section x.2 evaluates DQRM on the Kaggle dataset and on the Terabyte dataset under the distributed settings. Section x.3 evaluates DQRM's effect on communication

[*] Both authors contributed equally to this research.

Conference acronym 'XX, June 03–05, 2018, Woodstock, NY

ACM ISBN 978-1-4503-XXXX-X/18/06…$15.00
https://doi.org/XXXXXXX.XXXXXXX

compression over multiple GPUs and multiple node CPU server environment.

2.1 Quantization of Embedding Tables

It is an concensus in previous works ([1, 2]) that the embedding tables occupies over 99% of the entire DLRM model size. Therefore, compressing embedding tables is important to the overall DLRM inference model size compression. Experiment results presented in the section is evaluated on the Kaggle Dataset. We use 4 Nvidia M40 GPUs to do distributed DQRM QAT under only Data Parallelism (DP) settings. We deploy uniform quantization on the embedding tables only. We investigate three different bit width for quantizing Embedding Tables, INT16, INT8, and INT4. All three are trained in QAT for 3 epochs. Figure 1(a) shows the curves of the testing accuracy during 3 epochs of QAT using different quantization bit widths. The vertical dashed lines signifies the boundaries of each epoch. Through the experiments, we found that DLRM in single precision (blue) suffers from severe overfitting after the first epoch, while testing accuracies of INT16 (orange) and INT8 (green) quantization resembles the trend. In contrast, INT4 quantization (red), though converge slower at the beginning, overcomes overfitting and continue to rise over the entire five epochs. We report the model performance in 1. After 5 epochs of QAT, bit widths of INT16 and INT8 leads to comparable performance to the original unquantized DLRM model. Uniform INT4 quantization outperforms the original model in testing accuracy by 0.2% and ROC AUC score by 0.0047.

Table 1: DLRM Embedding Tables Quantization, accuracies evaluated on the Kaggle Dataset

Quantization	Testing	
Bit Width	Accuracy	ROC AUC
Unquantized	78.729%	0.7993
INT16	78.772% (+0.043%)	0.8005 (+0.0012)
INT8	78.781% (+0.052%)	0.8007 (+0.0014)
INT4	**78.936% (+0.20%)**	**0.8040 (+0.0047)**

2.2 Quantization of the Whole Model

DQRM is based on DLRM which contains two major components in its network architecture, Embedding Table and Multi-Layer Perceptron (MLP). When adding the MLP layers quantization to DQRM, we observe that MLP models are more skeptical to quantization, and doing it naively leads to significant accuracy loss, same as in

Table 2: DQRM 4-bit quantization results evaluated on Kaggle and Criteo datasets

(a) 4-bit quantization results for DLRM on Kaggle

Quantization	Model	Training	Testing	
Settings	Bit Width	loss	Accuracy	ROC AUC
baseline	FP32	0.303685	78.718 %	0.8001
vanilla PTQ	INT4	-	-	-
DQRM	INT4	0.436685	78.897 %	0.8035

(b) 4-bit quantization results for DLRM on Criteo

Quantization	Model	Training	Testing	
Settings	Bit Width	loss	Accuracy	ROC AUC
baseline	FP32	0.347071	81.165%	0.8004
vanilla PTQ	INT4	-	-	-
DQRM	INT4	0.412979	81.159%	0.7998

Table 3: Multi-node Experiment Results with 8-bit gradients, loss evaluated on the Terabyte Dataset

#Node	Training	Testing	
	Loss Drop	Acc Drop	ROC AUC Drop
2	-	-	-
4	-	-	-
8	-	-	-

ACKNOWLEDGMENTS

To Robert, for the bagels and explaining CMYK and color spaces.

REFERENCES

[1] Antonio Ginart, Maxim Naumov, Dheevatsa Mudigere, Jiyan Yang, and James Zou. 2019. Mixed Dimension Embeddings with Application to Memory-Efficient Recommendation Systems. *CoRR* abs/1909.11810 (2019). https://arxiv.org/abs/1909.11810

[2] Udit Gupta, Xiaodong Wang, Maxim Naumov, Carole-Jean Wu, Brandon Reagen, David Brooks, Bradford Cottel, Kim M. Hazelwood, Bill Jia, Hsien-Hsin S. Lee, Andrey Malevich, Dheevatsa Mudigere, Mikhail Smelyanskiy, Liang Xiong, and Xuan Zhang. 2019. The Architectural Implications of Facebook's DNN-based Personalized Recommendation. *CoRR* abs/1906.03109 (2019). https://arxiv.org/abs/1906.03109

[3] Maxim Naumov, Dheevatsa Mudigere, Hao-Jun Michael Shi, Jianyu Huang, Narayanan Sundaraman, Jongsoo Park, Xiaodong Wang, Udit Gupta, Carole-Jean Wu, Alisson G. Azzolini, Dmytro Dzhulgakov, Andrey Mallevich, Ilia Cherniavskii, Yinghai Lu, Raghuraman Krishnamoorthi, Ansha Yu, Volodymyr Kondratenko, Stephanie Pereira, Xianjie Chen, Wenlin Chen, Vijay Rao, Bill Jia, Liang Xiong, and Misha Smelyanskiy. 2019. Deep Learning Recommendation Model for Personalization and Recommendation Systems. *CoRR* abs/1906.00091 (2019). https://arxiv.org/abs/1906.00091

A RESEARCH METHODS

A.1 Part One

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Morbi malesuada, quam in pulvinar varius, metus nunc fermentum urna, id sollicitudin purus odio sit amet enim. Aliquam ullamcorper eu ipsum vel mollis. Curabitur quis dictum nisl. Phasellus vel semper risus, et lacinia dolor. Integer ultricies commodo sem nec semper.

A.2 Part Two

Etiam commodo feugiat nisl pulvinar pellentesque. Etiam auctor sodales ligula, non varius nibh pulvinar semper. Suspendisse nec lectus non ipsum convallis congue hendrerit vitae sapien. Donec

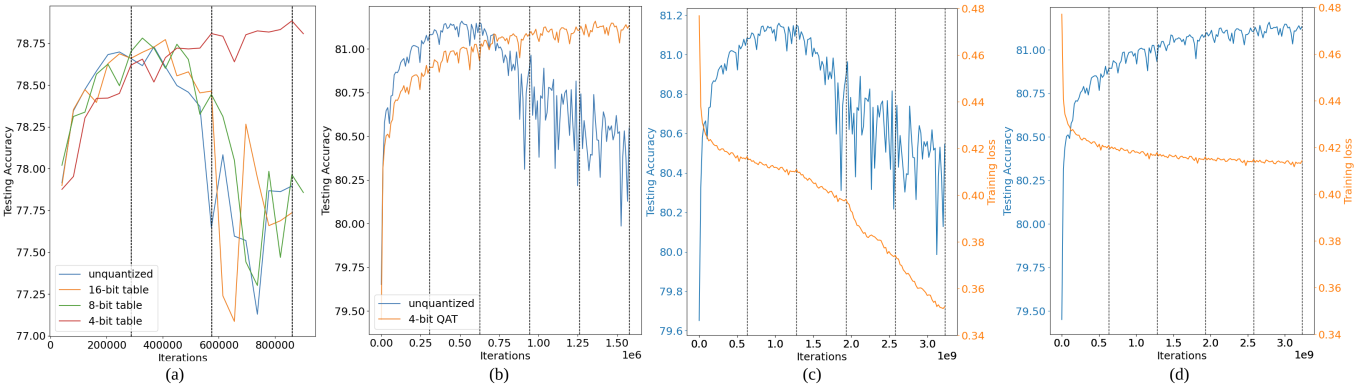


Figure 1: (a) shows the effect of different QAT bitwidths has on embedding tables in DLRM for three epochs of training (epochs are separated by black dashed line in all figure). QAT in uniform 4-bit overcomes the severe overfitting suffered by the original DLRM training and lead to significantly higher testing accuracy over three epochs of training. (b) shows the comparison between QAT in 4-bit for DLRM compared to normal training on the Terabyte dataset, and QAT in 4-bit leads to negligible accuracy drop while successfully overcoming the overfitting problem. (c) shows that the training loss (orange curve) for normal training starts decreasing drastically in the third epoch, right where the overfitting occurs. In (d), the training loss curve for QAT in 4-bit decreases stably throughout five epochs of training.

Table 4: Comparison of communication compression techniques for distributed training of 4 GPUs on the Kaggle Dataset

Communication Compression settings	Communication Overhead per iter	Latency per iter	Testing Accuracy	ROC AUC
gradient uncompressed	-	-	-	-
emb gradient sparfication (specified) [1]	-	-	-	-
emb gradient sparfication + 8-bit quantization	-	-	-	-

Table 5: Evaluation of Periodic Update on Kaggle and Terabyte Datasets

Model Type	Period	Latency per iter	Testing Accuracy	ROC AUC
Kaggle	1	-	-	-
	200	-	-	-
	500	-	-	-
Terabyte	1	-	-	-
	200	-	-	-
	500	-	-	-
	1000	-	-	-

at laoreet eros. Vivamus non purus placerat, scelerisque diam eu, cursus ante. Etiam aliquam tortor auctor efficitur mattis.

B ONLINE RESOURCES

Nam id fermentum dui. Suspendisse sagittis tortor a nulla mollis, in pulvinar ex pretium. Sed interdum orci quis metus euismod, et sagittis enim maximus. Vestibulum gravida massa ut felis suscipit congue. Quisque mattis elit a risus ultrices commodo venenatis eget dui. Etiam sagittis eleifend elementum.

Nam interdum magna at lectus dignissim, ac dignissim lorem rhoncus. Maecenas eu arcu ac neque placerat aliquam. Nunc pulvinar massa et mattis lacinia.

Received 20 February 2007; revised 12 March 2009; accepted 5 June 2009

Table 6: Evaluation of Gradient Quantization Bit Width

Embedding Gradient Bit Width	Communication Overhead per iter	Testing Accuracy	ROC AUC
Unquantized	-	-	-
INT16	-	-	-
INT8	-	-	-
INT4	-	-	-

Table 7: Quantization Evaluation of Each Part of the Model

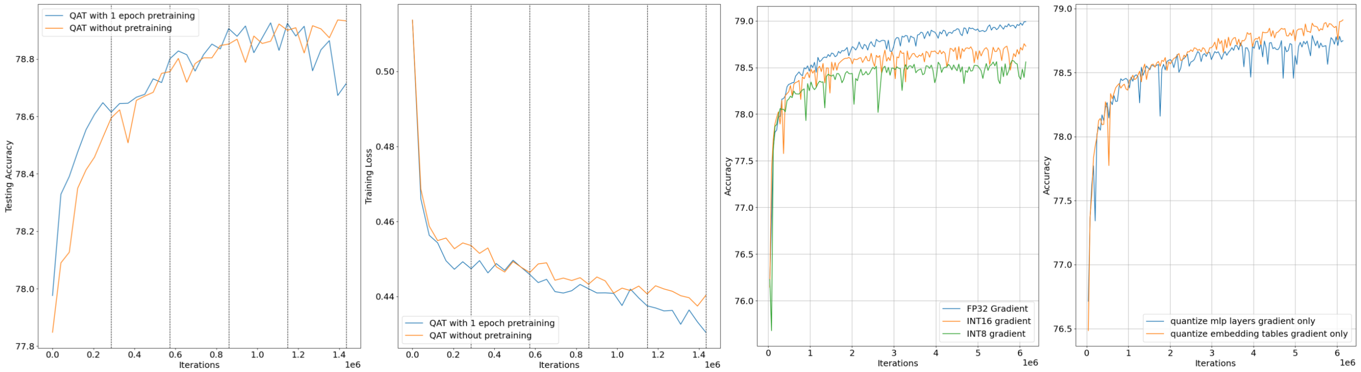
Settings	Testing Accuracy	Testing ROC AUC
Baseline	-	-
+ Embedding Tables in 4-bit	-	-
+ MLP in 4-bit matrix-wise	-	-
+ MLP in 4-bit channelwise	-	-

Table 8: Quantization Evaluation of Each Part of the Model

Settings	Testing Accuracy	Testing ROC AUC
Baseline	-	-
+ Embedding Tables in 4-bit	-	-
+ MLP in 4-bit matrix-wise	-	-
+ MLP in 4-bit channelwise	-	-

Table 9: Evaluation of QAT as a Finetuning Technique

Settings	Testing Accuracy	Testing ROC AUC
One epoch pretraining + Four epochs of QAT	-	-
Five epochs of QAT without pretraining	-	-



(a) (b) (c) (d)

Figure 2: (a) Testing accuracy over the 5 epochs for 1 epoch of pretraining before 4 epochs of QAT and 5 epoch of QAT without pretraining. Pretraining leads to faster overfitting, and QAT without pretraining avoid overfitting and achieves better testing accuracy from 5 epochs of training. (b) Training loss over the 5 epochs. Pretraining before QAT leads to a overall faster decrease of training loss in DLRM compared with QAT without pretraining. (c) Testing Accuracy of naively quantizing gradients for communication into different bit widths. Naive gradient quantization leads to significant accuracy drop. (d) MLP gradients are more sensitive to quantization. If only quantize embedding table gradients, it will leads to less drop in accuracy compared to only quantizing MLP gradients.