로그 양자화 기법과 고정소수점 변환을 이용한 GRU 네트워크 최적화 방법

박상기, 정기석*
한양대학교

pskhanyang@gmail.com, *kchung@hanyang.ac.kr

Optimizing GRU Networks using Log-quantization and Fixed-point Conversion

Sangki Park, Ki-Seok Chung*
Hanyang University, Seoul, Korea

요 약

Recurrent neural network 은 최근 정확도를 높이기 위하여 네트워크 크기가 커지는 경향을 보여주고 있다. 하지만 큰 네트워크를 양자화 없이 그대로 사용하는 것은 매우 고성능의 연산 능력을 요구하게 된다. 따라서, 본 논문은 로그 양자화와 고정 소수점 변환을 통해서 네트워크의 연산 복잡도를 줄이는 기법을 제안한다. Recurrent neural network 구조의 일종인 gated recurrent unit 을 기반한 네트워크에서, 기존의 미리 학습된 가중치를 추가 학습 없이도 로그 양자화 기법을 적용한 후에도 정확도를 유지할 수 있음을 보인다. 실험을 통해 로그 양자화 및 고정소수점 변환 기법은 부동소수점 방식 대비 메모리 사용량이 68%로 줄어들고, 표현 bit 수에 따른 정확도 감소는 2.6%로 작다는 것을 보인다.

Ⅰ. 서 론

Recurrent neural network (RNN)은 convolutional neural network (CNN)과는 다르게 순서가 있는 데이터 처리에 사용된다. RNN 은 기계어 번역, 음성 인식, image captioning 등의 분야에서 좋은 결과를 보여주고 있다 [1,2]. 하지만 RNN 은 시간에 대한 의존성 때문에 CNN 에 비해 병렬성이 떨어지고 정확도를 올리기 위해 네트워크의 크기 또한 매우 큰 경우가 많다. 특히 [1]의 경우, 큰 모델이 24 개의 레이어, 전체 웨이트의 개수가 3 억 4 천만개에 이른다. 이런 경우 메모리의 사용량이 증가할 뿐만 아니라 연산 자원이 많이 필요하다. 따라서 연산 속도 증가, 메모리 사용량 및 전력 소모량 감소를 위해서는 RNN 모델의 경량화 및 최적화가 필수적이다.

이미 CNN 모델에서는 최적화 기법이 널리 사용된다. 이 논문에서 주로 사용하는 양자화 기법에 대한 연구도 많이 진행되었다 [3,4]. 이 중 하나인 deep compression [3]에서는 양자화 이외에도 pruning 과 엔트로피 코딩을 이용해 VGG-16 네트워크 사이즈를 552 MB 에서 11.3 MB 으로 49 배 압축하였음에도 정확도 감소는 Top-5 기준으로 0.41%에 불과하다는 것을 보였다. [4]의 경우 로그 양자화 기법으로 fully-connected 층은 4-bit, convolution 층은 5-bit 만을 사용하였고, 정확도 감소는 Top-5 에서 1.7%였다.

본 논문에서는 로그 양자화 기법과 고정소수점 변환을 RNN 구조중의 하나인 gated recurrent unit (GRU) [5] 기반 네트워크에 적용하여 네트워크 크기와 메모리 사용량을 줄이는 방법을 제시한다. 32-bit 부동소수점 (floating-point, FP)의 기존 방식에 비해 최대 8-bit 까지 표현 비트를 줄이며, 로그 양자화를 적용하면, 곱셈 대신 shift 와 덧셈 연산을 사용하여 계산을 할 수 있다.

Ⅱ. 본론

2.1 RNN 의 구조

RNN 모델의 레이어는 여러 개의 셀로 이루어진다. 이때 한 개의 RNN 셀에서는 현재 시간의 입력 값 x_t 와 이전 시간의 결과값 h_{t-1} 을 이용해 현재 시간의 결과값 h_t 을 계산한다. 이때 h_t 는 시간 t 에 대한 hidden state 라고 하고, 다음과 같이 계산된다.

$$h_t = tanh(\boldsymbol{W}_h h_{t-1} + \boldsymbol{W}_x x_t + b) \quad (1)$$

여기서 $\boldsymbol{W}$ 는 가중치 행렬, b 는 bias 벡터이며 하나의 레이어 내 모든 셀에서 공유된다. 다만 위와 같은 단순 RNN 셀의 경우 시간이 길어지면 학습이 어려워지는 vanishing gradient 문제가 발생한다. 이를 해결하기 위해 long-short term memory (LSTM)와 GRU 구조가 대신 사용된다. GRU 의 hidden state 계산은 아래와 같다.

$$\begin{aligned} r_t &= \sigma(W_{xr}x_t + W_{hr}h_{t-1} + b_r) \\ z_t &= \sigma(W_{xz}x_t + W_{hz}h_{t-1} + b_z) \\ \tilde{h}_t &= tanh(W_{xh}x_t + W_{hh}(h_{t-1} \odot r_t) + b_h) \\ h_t &= z_t \odot h_{t-1} + (1 - z_t) \odot \tilde{h}_t \end{aligned} \quad (2)$$

2.2 양자화 기법

본 논문에서는 [4]와 비슷한 방식으로 32-bit FP 를 8-bit 고정 소수점 로그 형태로 변환한다. IEEE 754 의 단일 정밀도 부동소수점은 그림 1 과 같은 양식을 가진다. 여기서 부호 비트는 그대로 가져오고, 8-bit 지수 부분에서 127 바이어스를 뺀 후, MSB 를 제외한 7 비트를 signed integer 형식으로 가져온다. 이때 23-bit 의 가수부분은 지수부분을 반올림할 때 사용하게 된다. 이를 통하여, 32-bit 부동 소수점과의 곱셈을 지수 부분 간의 덧셈으로 대체할 수 있다.

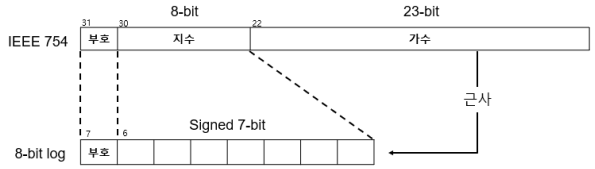

그림 1. 8-bit 형식 로그 양자화 방법

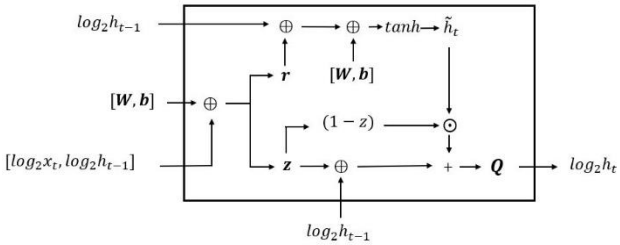

그림 2. 변형된 GRU 셀 구조

다음과 같은 변형을 거친 GRU 셀의 도식도는 그림 2 와 같다. 실험에서는 GRU 의 입력 벡터 x_t와 hidden state h_t를 양자화 하였고, 가중치와 바이어스는 32-bit 부동소수점을 사용하였다. 성능 비교를 위해 32-bit 부동소수점과 16 비트, 8 비트 고정소수점의 세 가지 정밀도를 가지는 모델을 추가로 구성해, 정확도와 메모리 사용량, 실행 시간을 측정하였다.

2.3 실험 및 결과

실험에 사용된 네트워크는 C 로 작성되었으며 64 개의 셀을 가지는 한 개의 GRU 층, 한 개의 fully-connected 층으로 구성 되어있다. 데이터 셋으로는 Shakespeare [6]을 사용하였으며 가중치는 Tensorflow 를 이용해 미리 학습된 것을 추가 학습 없이 데이터의 형식만 바꿔 이용하였다.

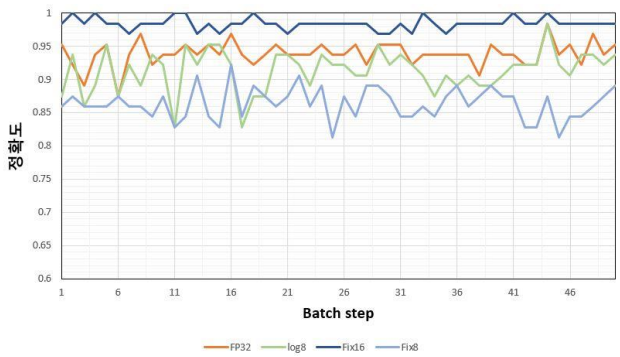

그림 3. 테스트 정확도 곡선 그래프

Model	FP32	Log8	Fix16	Fix8
평균 정확도	0.939	0.913	0.984	0.863
실행시간 (ms)	18.588	23.304	63.836	58.787
실행시간 비율	1	1.254	3.434	3.163
메모리 사용량 (KB)	30.157	20.595	15.091	7.558
메모리사용량 비율	1	0.683	0.500	0.251

표 1. 정밀도 별 정확도, 실행시간, 메모리 사용량

표 1 의 실험 결과, 정확도 면에서 가장 높은 것은 16 비트 고정 소수점에서의 98.4%였다. 본 논문에서 제안한 로그 양자화 방식은 91.3%로 32 비트 부동소수점 (FP32) 사용시 보다는 2.6% 낮고, 8 비트 고정 소수점보다는 5% 높은 정확도를 보여주었다. 또한 메모리 사용량 측면에서, 로그양자화와 고정소수점 변환 방식은 큰 정확도 감소 없이 메모리 사용량을 FP32 방식 대비 68%만 사용한다. 로그 양자화와 고정소수점 변환 방식에서는 FP32 데이터를 변환하는 함수와 고정소수점을 위한 곱셈 함수가 포함되어 있어 FP32 에 비해 실제 실행시간은 늘어나나, 이 점은 구현 언어인 C 의 비트 연산이 자유롭지 않은 점에서 기인하므로, Verilog HDL 등의 언어로 설계가 기술된다면 훨씬 큰 성능 증가를 기대할 수 있다.

Ⅲ. 결론

본논문에서는 로그 양자화 기법이 GRU 네트워크에 적용하여 기존 가중치를 추가 학습 없이 바로 적용 가능하며, 이에 따른 정확도 감소는 2.6%에 불과함을 보였다. 메모리 사용량은 32 비트 부동소수점 표현 대비 68%로 줄어든다. 추후, Verilog HDL 로 설계를 구현하면 C 의 비효율적인 bit 연산 방식을 벗어날 수 있으므로 더욱 높은 성능 향상을 기대할 수 있을 것이다.

ACKNOWLEDGMENT

본 연구는 IDEC 에서 EDA Tool 를 지원받아 수행하였습니다.

참 고 문 헌

[1] Jacob Devlin, Ming-Wei Chang, et al. "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding," arXiv:1810.04805, 2018.

[2] Quanzeng You, Hailin Jin, et al. "Image Captioning with Semantic Attention," The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016, pp. 4651-4659.

[3] Song Han, Huizi Mao, et al. “Deep Compression: Compressing Deep Neural Networks with Pruning, Trained Quantization and Huffman Coding,” International Conference on Learning Representations, May. 2016

[4] Daisuke Miyashita, Edward H. Lee, et al. “Convolutional Neural Networks using Logarithmic Data Representation,” arXiv:1603.01025, 2016.

[5] Cho Kyunghyun, Bart van Merrienboer, et al. “Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation,” Conference on Empirical Methods in Natural Language Processing (EMNLP), pp. 1724-1734, 2014.

[6] William Shakespeare Plays Datasets, https://ocw.mit.edu/ans7870/6/6.006/s08/lecturenotes/files/t8.shakespeare.txt