ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) | 978-1-7281-7605-5/20/$31.00 ©2021 IEEE | DOI: 10.1109/ICASSP39728.2021.9414076

MIXED PRECISION QUANTIZATION OF TRANSFORMER LANGUAGE MODELS FOR SPEECH RECOGNITION

Junhao Xu, Shoukang Hu, Jianwei Yu, Xunying Liu, Helen Meng

The Chinese University of Hong Kong, Hong Kong SAR, China

ABSTRACT

State-of-the-art neural language models represented by Transformers are becoming increasingly complex and expensive for practical applications. Low-bit deep neural network quantization techniques provides a powerful solution to dramatically reduce their model size. Current low-bit quantization methods are based on uniform precision and fail to account for the varying performance sensitivity at different parts of the system to quantization errors. To this end, novel mixed precision DNN quantization methods are proposed in this paper. The optimal local precision settings are automatically learned using two techniques. The first is based on a quantization sensitivity metric in the form of Hessian trace weighted quantization perturbation. The second is based on mixed precision Transformer architecture search. Alternating direction methods of multipliers (ADMM) are used to efficiently train mixed precision quantized DNN systems. Experiments conducted on Penn Treebank (PTB) and a Switchboard corpus trained LF-MMI TDNN system suggest the proposed mixed precision Transformer quantization techniques achieved model size compression ratios of up to 16 times over the full precision baseline with no recognition performance degradation. When being used to compress a larger full precision Transformer LM with more layers, overall word error rate (WER) reductions up to 1.7% absolute (18% relative) were obtained.

Index Terms— Language models, Speech recognition, Transformer, Quantization, ADMM

1. INTRODUCTION

Deep Transformer models in recent years have defined state-of-the-art language modelling performance across a range of applications including automatic speech recognition (ASR). The Transformer model architecture features a deep stacking of multiple self-attention layers [1, 2, 3] with residual connections [4] and layer normalization [5] Additional positional encoding layers [6, 7] can be used to further augment the self-attention layers with sequence order information. Performance improvements over the conventional long short-term memory recurrent neural network (LSTM-RNN) language models have been widely reported [8, 9]. However, the deeper architecture design of Transformers not only leads to a large increase in overall system complexity, memory footprint and computational cost when operating on the cloud, but also creates difficulty when deploying them on edge devices to enhance privacy and reduce latency, in common with many other computational intensive deep learning applications that are currently facing similar challenges.

To this end, one powerful solution recently drawing increasing interest in the machine learning and speech technology community is to use low-bit deep neural network (DNN) quantization techniques [10, 11]. By replacing floating point weights with low precision values, for example, binary numbers, quantization can dramatically reduce the model size without modifying the network architecture [12, 13, 14]. Further model size reduction can be obtained when low-precision quantization is used in combination with neural architecture search (NAS) methods, for example, in the SqueezeNet system [15]. In contrast to the extensive research works on low-bit quantization methods conducted on computer vision tasks [16, 17], only limited previous research in this direction has been conducted in the context of language modelling [18] and ASR systems.

Two issues are associated with current low-bit DNN quantization methods. First, these quantization approaches are predominantly based on uniform precision, where an identical bit-width is applied to all weight parameters for quantization. This fails to account for the varying performance sensitivity at different parts of the system to quantization errors. In practice, this often leads to large performance degradation against full precision models. Second, gradient descent methods and back-propagation (BP) algorithm cannot be directly applied in quantized model training when the weights are restricted to discrete values. Existing methods of training low-bit quantized DNNs often use a modified BP algorithm [19, 20], where low precision quantized parameters were first used in the forward pass to compute the error loss before full precision parameters are used in the backward pass to propagate the gradients for model update. However, the direct use and estimation of quantized weights in these methods leads to very slow convergence in training, while the performance gap against full precision models remains.

In order to address these issues, novel mixed precision DNN quantization methods are proposed in this paper to address this problem by applying locally variable bit-widths to individual components of the system. These methods are becoming well supported by the recent development of mixed precision DNN acceleration hardware that allow multiple locally set precision settings to be used [21]. The resulting flexibility can provide a better trade-off between compression ratio and accuracy performance target. The optimal local precision settings are automatically learned using two techniques. The first is based on a quantization sensitivity metric in the form of Hessian trace weighted quantization perturbation. It can be efficiently computed using Hessian-free approaches. The second is based on mixed precision Transformer architecture search.

In order to overcome the difficulty in using gradient descent methods to directly estimate DNNs of discrete quantized weights, alternating direction methods of multipliers (ADMM) are proposed to efficiently train mixed precision quantized DNN systems. Experiments conducted on multiple tasks: Penn Treebank (PTB), Switchboard (SWBD) suggest the proposed mixed precision Transformer LM quantization techniques achieved a model size compression ratio of up to 16 times over the full precision baseline with no recognition performance degradation. Moreover, by applying quantization to a more complex Transformer LM with more layers, we can get overall WER reduction up to 1.7% absolute.

The main contributions of this paper are summarized as following. First, to the best of our knowledge, this paper is the first

work to apply mixed precision quantization methods to Transformer language models. In contrast, previous researches on low-bit quantization focused on convolutional neural networks (CNNs) [22] and LSTM-RNN LMs [23], where expert designed special partially quantized linear layers containing binary weight matrices, full precision bias and additional scaling parameters were used mitigate the performance degradation due to uniform precision quantization.

The rest of the paper is organized as follows. Transformer LMs are reviewed in section 2. A general neural network quantization scheme and uniform quantization are presented in section 3. Section 4 presents our mixed precision quantization methods in details. Experiments and results are shown in section 5. Finally, conclusions and future work are discussed in section 6.

2. TRANSFORMER LMS

The Transformer model architecture considered in this paper feature a deep stacking of multihead attention followed by feedforward layers. Residual connections and layer normalization are also inserted between them as in the top part of figure 1. The l-th Transformer layer transforms the input $\mathbf{x^{l-1}}$ at t time step as follows:

$$\mathbf{q}_t^l, \mathbf{k}_t^l, \mathbf{v}_t^l = \mathbf{Q}\mathbf{x}_t^{l-1}, \mathbf{K}\mathbf{x}_t^{l-1}, \mathbf{V}\mathbf{x}_t^{l-1} \quad (1)$$

$$\mathbf{h}_t^l = (\mathbf{h}_{t-1}^l, (\mathbf{k}_t^l, \mathbf{v}_t^l)) \quad (2)$$

$$\mathbf{y}_t^l = \mathbf{W}_h^l \text{SelfAttention}(\mathbf{h}_t^l, \mathbf{q}_t^l) + \mathbf{x}_t^{l-1} \quad (3)$$

$$\mathbf{z}_t^l = \text{LayerNorm}(\mathbf{y}_t^l) \quad (4)$$

where $\mathbf{Q}, \mathbf{K}, \mathbf{V}$ are the query, key, value projection matrices. $\mathbf{h^l}$ stores the history information in the l-th self-attention layer. $(,)$ denotes vector concatenation operation. SelfAttention$(\cdot)$ is the scaled multi-head dot product self-attention machanism and $\mathbf{W_h^l}$is the projection matrix. LayerNorm$(\cdot)$ is the layer normalization.

The feed-forward layer at time step t is:

$$\mathbf{s}_t^l = \mathbf{W}_2^l \text{GELU}(\mathbf{W}_1^l \mathbf{z}_t^l + \mathbf{b_1^l}) + \mathbf{b_2^l} + \mathbf{z_t^l} \quad (5)$$

$$\mathbf{x}_t^l = \text{LayerNorm}(\mathbf{s}_t^l) \quad (6)$$

where $\mathbf{W_1^l}$ and $\mathbf{W_2^l}$ are the weight matrices and $\mathbf{b_1^l}$ and $\mathbf{b_2^l}$ are the corresponding bias. GELU$(\cdot)$ represents the Gaussian error linear unit [24]. In addition, we also use positional embedding layer in the transformer LMs.

3. NEURAL NETWORK QUANTIZATION

For a standard n-bit quantization problem of neural networks, we consider a full precision weight parameter θ and find its closest discrete approximation from the following quantization table $q \in \{0, \pm 1, \pm 2, \ldots, \pm(2^{n-1}-1)\}$ as

$$f(\theta) = \arg\min_q |\theta - q| \quad (7)$$

1 bit is reserved to denote sign bit. With further simplification, extremely low bit quantization, for example, binarization $\{1, -1\}$ [25, 26] and ternary $\{-1, 0, 1\}$ [27], can be produced.

Applying quantization to all weight matrices in the model, we can use a more general format in equation (7) to represent the quantization for each parameter. Let $\theta_i^{(l)}$ be the i^{th} parameter within any of the l^{th} weight cluster, for example, all weight parameters of the same layer,

$$f(\theta_i^{(l)}) = \arg\min_{Q_i^{(l)}} |\theta_i^{(l)} - Q_i^{(l)}| \quad (8)$$

The locally shared l^{th} quantization table is given by

$$Q_i^{(l)} \in \{0, \alpha^{(l)}, \ldots, \alpha^{(l)}(2^{n-1}-1)\} \quad (9)$$

where $\alpha^{(l)}$ is a full precision scaling factor used to adjust the dynamic range of all the unquantized weights in the cluster. It is shared locally among weight parameters clusters. A special case, when the local quantization table in equation (8) is shared across all the layers, leads to the traditional uniform precision quantization approach. The only remaining factor affecting the system performance is the bit length $\#bit$ which is also globally set to be $1, 2, 4, 8$ etc.

4. MIXED PRECISION TRANSFORMER QUANTIZATION

This section presents three mixed precision based Transformer LM quantization approaches.

4.1. ADMM Based Mixed Precision Quantization

One major challenge faced by both uniform and mixed precision quantization is that the gradient descent methods and back-propogation (BP) algorithm can not be directly used when weights are quantized to discrete values. To this end, mixed precision BP was proposed later in [20] where low precision binarized parameters were first used in the forward pass to compute the error loss before full precision parameters are used in the backward pass to propagate the gradients. However, directly training quantized system using mixed precision BP leads to very slow convergence and the performance gap between full precision and quantized systems remains large. An alternative solution to this problem is to reformulate quantization as a constrained optimization problem implemented solved by the alternating direction methods of multipliers (ADMM) [28]. It was initially used to in [29] learn the global quantization table in equation (8) where α is shared among all the parameters.

In order to account for the locally varying performance sensitivity, ADMM was used in our earlier research [30] to learn the local quantization tables in equation (8). This allows ADMM to provide a form of mixed precision quantization. However, the optimum local quantization precision settings cannot be learned by ADMM and must be manually set. These will be automatically learned in the following two approaches of Sections 4.2 and 4.3.

4.2. Minimum Sensitivity Based Mixed Precision Quantization

Assuming the parameters of a neural network is twice differentiable and converged to a local optimum, it was proved in [16] that the expected performance loss, when using a given quantization precision, is expressed in the form of Hessian trace weighted squared quantization error. In simple terms, for each cluster of weight parameters, given the same amount of weight perturbation resulted from quantization, the smaller the associated Hessian matrix trace, the lower the performance sensitivity to quantization.
For any quantization $\mathbf{Q}(\cdot)$ being applied to the network parameters $\mathbf{W}$, the total performance sensitivity can be represented by the following sum of Hessian trace and squared quantization perturbation error.

$$\Omega = \sum_{i=1}^{L} \Omega_i = \sum_{i=1}^{L} \bar{Tr}(\mathbf{H_i}) \cdot ||\mathbf{Q}(\mathbf{W_i}) - \mathbf{W_i}||_\mathbf{2}^\mathbf{2} \quad (10)$$

Given a target average quantization precision, the local quantization bit widths used in each layer should be selected such that the above total performance sensitivity is minimized. In practice, this requires transformer LMs using uniform precision, for example 1-bit, 2-bit, 4-bit and 8-bit be separately trained off-line first via ADMM

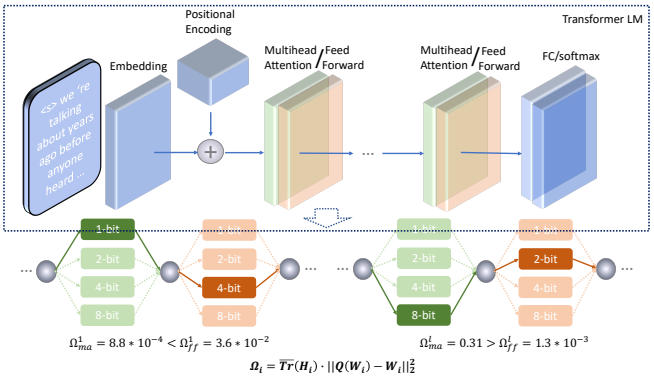

Fig. 1. *An example of auto-configured mixed precision quantization of a transformer LM using a minimum performance sensitivity measure. For the first transformer module positioned right after the embedding and position encoding layer, its multi-head attention layer (green) is uses binary quantization while its feed forward layer (orange) uses 4-bit quantization precision, as determinted by the Hessian-trace weighted quantization sensitivity measure.*

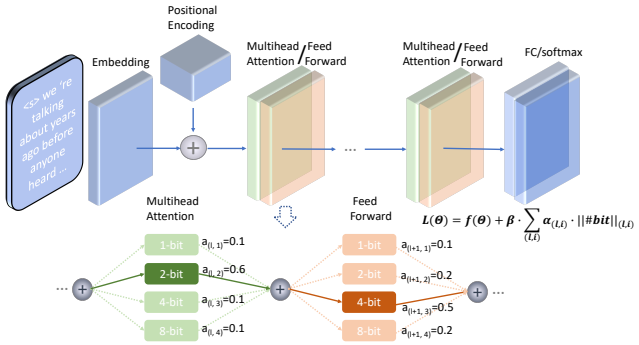

Fig. 2. *An example of auto-configured mixed precision quantization of a transformer LM using mixed precision architecture search. For the first transformer module, its multi-head attention layer is uses 2-bit quantization (green) given the associated selection weight of 0.6 while its feed forward layer uses 4-bit quantization precision (orange) given the associated selection weight of 0.5, as the 1-best choice selected from the mixed precision NAS super-network.*

optimization in Section 4.1. The performance sensitivity in equation (9) can then be computed locally for each layer using each quantization choices before taking the sum.

For larger transformer LMs containing millions of parameters, and many large deep neural networks in general, directly computing the Hessian matrix and its trace is infeasible. In order to handle this , an efficient stochastic linear algebra approach based on the Huchinson's Algorithm [31] is used to approximate the Hessian trace,

$$Tr(\mathbf{H}) \approx \frac{1}{\mathbf{m}} \sum_{\mathbf{i=1}}^{\mathbf{m}} \mathbf{z_i^T H z_i} \tag{11}$$

where the expensive matrix multiplication between $\mathbf{H}$ and $\mathbf{z_i}$ in the above approximation can be avoided, and efficiently computed using Hessian-free approaches [16]. $\mathbf{z_i}$ is a random vector sampled from a Gaussian Distribution $\mathcal{N}(\mathbf{0}, \mathbf{1})$.

An example application of the minimum performance sensitivity based mixed precision quantization of two layers within a Transformer LM is shown in figure 1 (residual connection and normallzation are omitted for brevity).

4.3. Architecture Search Based Mixed Precision Quantization

An alternative solution to automatically determine the suitable local quantization precision settings is to use mixed precision neural architecture search (NAS) [32]. Inside a NAS super-network containing all possible Transformer architectures with varying precision bit widths, the differentiable architecture weights [33] associated different precision settings can be automatically learned inside the super-network together with the normal Transformer parameters.

Instead of selecting over heterogeneous neural building structures as considered in conventional NAS applications, now for transformer LM quantization purposes, different neural building blocks, for example, Transformer modules of different bit-widths are considered. This major difference requires the associated mixed precision quantization super-network to be specially designed. Such super-network is constructed by first separately training transformer LMs using uniform precision, for example 1-bit, 2-bit, 4-bit and 8-bit, using ADMM optimization, before connecting these uniform precision quantized Transformer LMs at each layer, where the system specific activation outputs are linearly combined using a set of quantization precision selection weights ($\{\alpha_{(l,i)}\}$ in equation (12)). An example of such mixed precision Transformer super-network is shown in Figure 3.

In order to avoid the trivial selection of the longest, most generous quantization bit width, these precision selection weights learning can be further constrained by a model complexity penalty in terms of the number of bits retained after quantization. [1]

$$L(\boldsymbol{\theta}) = f(\boldsymbol{\theta}) + \beta \sum_{(i,l)} \boldsymbol{\alpha}_{(i,l)} \cdot \sqrt{||\#bit||_{(i,l)}} \tag{12}$$

where $f(\boldsymbol{\theta})$ is the standard cross-entropy loss.

5. EXPERIMENTS

In order to evaluate the performance of mixed precision quantized Transformer LMs, an initial set of experiments on the Penn Treebank (PTB) corpus are first presented in Section 5.1. The main set of experiments conducted on a Switchboard (SWBD) corpus are presented in Section 5.2. All mixture precision quantized Transformer LMs use layer level locally shared quantization tables with varying precision settings that are either manually in case of ADMM, or learned by minimum performance sensitivity (*MinSen*) and mixed precision neural architecture search (MPNAS) of Sections 4.2 and 4.3. Statistical significance test was conducted at level $\alpha = 0.05$ based on matched pairs sentencesegment word error (MAPSSWE) for recognition performance analysis.

5.1. Experiments on Penn Treebank Corpus

The PTB corpus uses a 10K word vocabulary. 930K words of text data were used for training. 74K and 82K words of development and test data sets were used.

There are several trend can be found in Table 2, given the same quantization precision, for example, at approximately 2 bits, all the mixed precision quantized models ADMM (line 7), MinSen (line 10) and MPNAS (line 11), consistently outperform the 2-bit uniform quantization model in line 3. Second, among all the mixed precision quantization methods, the lowest PPL of 56.82 is obtained using MinSen with a quantization ratio of 10.2 times.

[1] The square root of #bit provides a smoothing effect on the precision system complexity penalty term to avoid over-penalizing high precision settings

Table 1. *Performance of the baseline full precision, uniform precision quantized and layer level mixed precision quantized Transformer LMs with local precision set either manually in ADMM, or automatically using MinSen/MPNAS of Sections 4.2 & 4.3 on Switchboard NIST Hub5'00, RT02 and RT03. Evaluation time is computed over rescoring all the N-best lists.*

models	quant. precision	quant. method	#bit	PPL	+4gram PPL	Hub5'00 WER(%) rt02	rt03	swbd.	callhm.	model size(MB)	comp. ratio	evaluation time(s)
1			32	41.24	41.08	12.9	17.3	7.8	15.6	106	-	13.19
2			1	48.26	47.95	13.6	18.5	8.2	16.2	3.6	30.5	4.76
3		*uniform*	2	44.62	43.28	13.4	18.2	8.1	15.9	7.9	13.4	6.43
4		precision	4	43.83	42.97	13.2	17.8	8	15.7	14.1	7.5	6.89
5			8	43.72	41.36	13.0	17.4	7.9	15.8	27.2	3.9	7.12
6			1	47.26	46.10	13.5	18.3	8.1	16.1	3.6	30.5	4.76
7		ADMM	2	42.62	42.32	13.3	17.9	8.0	15.7	7.9	13.4	6.43
8	*mixed*		4	42.83	41.66	13.1	17.4	7.9	15.7	14.1	7.5	6.89
9	precision		8	42.72	41.21	13.0	17.4	7.8	15.8	27.2	3.9	7.12
10		MinSen	**1.9**	**42.39**	**41.52**	**13.0**	**17.5**	**7.9**	**15.7**	**8.0**	**13.25**	**6.58**
11		NAS.	2.5	42.75	41.96	13.2	17.8	7.9	15.8	9.1	11.65	6.80

Table 2. *Perplexity (PPL), quantization bit length #bit, model size and compression ratio of baseline full precision Transformer (*trans.*), uniform and mixed precision quantized Transformer using manual setting (*ADMM*), performance sensitivity based quantization (*MinSen*) and architecture search based quantization (*NAS*)*

models	quant. prec.	quant. meth.	#bit	PPL	Model. Size	Comp. Ratio
1			32	55.26	66	-
2			1	82.10	2.3	28.7
3	*uni.*	-	2	58.94	4.6	14.3
4	prec.		4	56.86	9.4	7.0
5			8	56.80	17.0	3.9
6			1	65.41	2.3	28.7
7		ADMM	2	58.06	4.6	14.3
8	*mixed*	(manual)	4	56.84	9.4	7.0
9	prec.		8	56.75	17.0	3.9
10		MinSen	**2.0**	**56.82**	**6.5**	**10.2**
11		NAS.	2.2	58.23	4.8	13.8

5.2. Experiments on Conversational Telephone Speech

The Switchboard I telephone speech corpus we use consists of approximately 300 hours of audio data released by LDC (LDC97S62). The baseline GMM-HMM system with 6008 tied tri-phone states was trained based on 40-dimensional Mel-frequency cepstral coefficients (MFCCs) to generate alignments for the neural network training. LF-MMI trained TDNN[34] acoustic models with data augmentation and i-Verctor adaptation [35] were used. Various Transformer LMs trained on the Switchboard and Fisher transcripts (LDC2004T19, LDC2005T19) was used to rescore the 4-gram LM[2] produced N-best lists ($N = 20$). Their performance are shown in Table 1.

Similar trends can be found in Table 1. First, given the same quantization precision, for example at approximately 2 bits, all the mixed precision quantized systems (ADMM, MinSen and MPNAS) outperform the equivalent 2-bit uniform quantized systems in line 3. Second, among the three mixed precision quantization approaches, auto-configured quantization by MinSen or MPNAS outperform the manual ADMM quantization using a comparable bit width of 2. In particular, the 1.9 bit quantized MinSen model (line 10) outperforms the comparable uniform precision (line 3) and manual ADMM quantization (line 7) by 0.4 to 0.7 and 0.3 to 0.4 absolution on rt02 and rt03 in a statistically significant manner. Finally, the evaluation time is also halved against full precision baseline.

The advantages of mixed precision quantization are further demonstrated when being used to compress a 16-layer larger Transformer LM (line2 in table 3), The 4-bit ADMM quantized model produced the best performance, with a statistically significant WER reductions of 1.7

[2]the same 4gram LM was used in both the initial lattice and N-best list generation stage, and subsequent N-best rescoring.

6. CONCLUSIONS

This paper presents a set of novel mixed precision based Transformer LM quantization techniques for the locally varying performance sensitivity to the use of low-bit precision during model compression. The optimal local precision settings are automatically learned by either minimizing the performance sensitivity, or mixed precision NAS. Experimental results conducted on state-of-the-art speech recognition tasks suggest the proposed mixed precision quantization methods outperform uniform precision based quantization, and can produce large model size compression ratios of up to 16 times over the full precision baseline with no performance degradation. Future research will focus on improving mixed precision quantization methods and their application to other ASR system components.

Table 3. *Performance of the baseline 6 or 16 layer full precision, and mixed precision quantized Transformer LMs with local precision set either manually in ADMM, or automatically using MinSen/MPNAS of Sections 4.2 & 4.3 on Switchboard NIST Hub5'00, RT02 and RT03.*

models	n_{layers}	quant. meth.	#bit	PPL	Hub5'00 WER(%) swbd.	callhm.	rt02	rt03
1	6	-	32	45.9	7.8	15.6	12.9	17.3
2	16	-	32	45.9	9.5	16.2	15.1	19.4
3			1	46.2	9.3	16.8	15.3	19.6
4		ADMM	2	45.7	8.1	16.0	14.7	18.7
5	16	(manual)	**4**	**45.1**	**7.8**	**15.7**	**13.5**	**17.3**
6			8	45.0	8.0	16.0	14.0	18.1
7		MinSen	2.8	45.3	7.8	15.8	13.9	17.7
8		NAS.	2.1	45.6	7.9	15.9	14.2	17.9

7. ACKNOWLEDGEMENT

This research is supported by Hong Kong RGC GRF grant No. 14200218, 14200220, TRS T45-407/19N, Innovation & Technology Fund grant No. ITS/254/19, and SHIAE grant No. MMT-p1-19.

8. REFERENCES

[1] J. Cheng, L. Dong, and M. Lapata, "Long short-term memory-networks for machine reading," in *EMNLP*. Austin TX USA, 2016, p. 551–561.

[2] Z. Lin, M. Feng, C. N. d. Santos, B. Xiang M. Yu, B. Zhou, and Y. Bengio, "A structured self-attentive sentence embedding," *Int. Conf. on Learning Representations (ICLR)*, Apr. 2017.

[3] A. P. Parikh, O. Ta ckstro m, D. Das, and J. Uszkoreit, "A decomposable attention model for natural language inference," in *EMNLP*. Austin TX USA, 2016, p. 2249–2255.

[4] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," *IEEE Conf. on Computer Vision and Patt. Recog. (CVPR)*, Jun. 2016.

[5] J.L.Ba, J.R.Kiros, and G.E.Hinton, "Layer normalization," in *arXiv preprint*. arXiv:1607.06450, 2016.

[6] JA. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," in *NIPS*. Long Beach CA, 2017, p. 5998–6008.

[7] J.Gehring, M.Auli, D.Grangier, D.Yarats, and Y.N.Dauphin, "Convolutional sequence to sequence learning," in *ICML*. Sydney Australia, 2017, p. 1243–1252.

[8] Kazuki Irie1, Albert Zeyer, Ralf Schlüter, and Hermann Ney, "Language modeling with deep transformers," *INTERSPEECH*, Sep. 2019.

[9] Albert Zeyer, Parnia Bahar, Kazuki Irie, Ralf Schlüter, and Hermann Ney, "A comparison of transformer and lstm encoder decoder models for asr," in *2019 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*. IEEE, 2019, pp. 8–15.

[10] K. Asanovic and N. Morgan, "Experimental determination of precision requirements for back-propagation training of artificial neural networks," *ICSI*, 1991.

[11] R. Meir D. Soudry, I. Hubara, "Expectation backpropagation: Parameter-free training of multilayer neural networks with continuous or discrete weights," *NIPS*, 2014.

[12] M. Rastegari, V. Ordonez, J. Redmon, and A. Farhadi, "Xnor-net: Imagenet classification using binary convolutional neural networks," *ECCV*, 2016.

[13] J. Wu et al, "Quantized convolutional neural networks for mobile devices," *CVPR*, 2016.

[14] A. Zhou, A. Yao, Y. Guo, and L. Xu et al, "Incremental network quantization: Towards lossless cnns with low-precision weights," *ICLR*, 2017.

[15] Forrest N Iandola, Song Han, Matthew W Moskewicz, Khalid Ashraf, William J Dally, and Kurt Keutzer, "Squeezenet: Alexnet-level accuracy with 50x fewer parameters and $<$ 0.5 mb model size," *arXiv preprint arXiv:1602.07360*, 2016.

[16] Z. Dong abd Z. Yao, A. Gholami, M. W. Mahoney, and K. Keutzer, "Hawq: Hessian aware quantization of neural networks with mixed-precision," *ICCV*, 2019.

[17] Z. Liu K. Wang and et al, "Haq: Hardware-aware automated quantization," *CVPR*, 2019.

[18] Yoshua Bengio, Réjean Ducharme, Pascal Vincent, and Christian Jauvin, "A neural probabilistic language model," *Journal of machine learning research*, vol. 3, no. Feb, pp. 1137–1155, 2003.

[19] Y. Bengio et al I. Hubara, M. Courbariaux, "Quantized neural networks: Training neural networks with low precision weights and activations," *JMLR*, 2017.

[20] J. David M. Courbariaux, Y. Bengio, "Binaryconnect: Training deep neural networks with binary weights during propagations," *NIPS*, 2015.

[21] K. Keutzer et al Z. Dong, M. W. Mahoney, "Hawq-v2: Hessian aware trace-weighted quantization of neural networks," *arxiv.org/abs/1911.03852*, 2019.

[22] X. Xiang Y. Qian, "Binary neural networks for speech recognition," *Frontiers Inf. Technol. Electronic Eng*, 2019.

[23] K. Yu, R. Ma, K. Shi, and Q. Liu, "Neural network language model compression with product quantization and soft binarization," *IEEE TASLP*, 2020.

[24] D.Hendrycks and K.Gimpel, "Gaussian error linear units (gelus)," *arXiv preprint arXiv:1606.08415*, 2018.

[25] Cong Leng, Zesheng Dou, Hao Li, Shenghuo Zhu, and Rong Jin, "Extremely low bit neural network: Squeeze the last bit out with admm," in *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

[26] Mohammad Rastegari, Vicente Ordonez, Joseph Redmon, and Ali Farhadi, "Xnor-net: Imagenet classification using binary convolutional neural networks," in *European Conference on Computer Vision*. Springer, 2016, pp. 525–542.

[27] Fengfu Li, Bo Zhang, and Bin Liu, "Ternary weight networks," *arXiv preprint arXiv:1605.04711*, 2016.

[28] Neal Parikh Stephen Boyd and Eric Chu, "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Now Publishers Inc*, 2011.

[29] Chen Xu, Jianqiang Yao, Zhouchen Lin, Wenwu Ou, Yuanbin Cao, Zhirong Wang, and Hongbin Zha, "Alternating multi-bit quantization for recurrent neural networks," *International Conference on Learning Representations*, 2018.

[30] Junhao Xu, Xie Chen, Shoukang Hu, Jianwei Yu, Xunying Liu, and Helen Meng, "Low-bit quantization of recurrent neural network language models using alternating direction methods of multipliers," *ICASSP*, 2020.

[31] Haim Avron and Sivan Toledo, "Randomized algorithms for estimating the trace of an implicit symmetric positive semi-definite matrix," *Journal of the ACM (JACM)*, vol. 58, no. 2, pp. 1–34, 2011.

[32] Chenxi Liu, Barret Zoph, Maxim Neumann, Jonathon Shlens, Wei Hua, Li-Jia Li, Li Fei-Fei, Alan Yuille, Jonathan Huang, and Kevin Murphy, "Progressive neural architecture search," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 19–34.

[33] Hanxiao Liu, Karen Simonyan, and Yiming Yang, "Darts: Differentiable architecture search," in *International Conference on Learning Representations*, 2018.

[34] Daniel Galvez Daniel Povey, Vijayaditya Peddinti and et al, "Purely sequence-trained neural networks for asr based on lattice-free mmi.," in *Interspeech*, 2016.

[35] Najim Dehak, Patrick J Kenny, Réda Dehak, Pierre Dumouchel, and Pierre Ouellet, "Front-end factor analysis for speaker verification," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 19, no. 4, pp. 788–798, 2011.