Zero-Shot Dynamic Quantization for Transformer Inference

Yousef El-Kurdi* **Jerry Quinn*** **Avirup Sil**
IBM Research AI
{yousefelk, jlquinn, avi}@us.ibm.com

Abstract

We introduce a novel run-time method for significantly reducing the accuracy loss associated with quantizing BERT-like models to 8-bit integers. Existing methods for quantizing models either modify the training procedure, or they require an additional calibration step to adjust parameters that also requires a selected held-out dataset. Our method permits taking advantage of quantization without the need for these adjustments. We present results on several NLP tasks demonstrating the usefulness of this technique.

1 Introduction

Transformer-based Neural Networks (NN) such as BERT (Devlin et al., 2018), RoBERTa (Liu et al., 2019) and XLM-R (Conneau et al., 2019), pre-trained on large amounts of data, have led to state-of-the-art (SOTA) results on many NLP tasks such as machine translation (Zhu et al., 2019), text classification (Wang et al., 2018) and question answering (Kwiatkowski et al., 2019; Clark et al., 2020). However, run-time inference of such large models is very costly due to their large computational requirements. In addition, deploying these models on smaller footprint mobile devices (Ravi and Kozareva, 2021) or cost-effective (Sanh et al., 2019; Jiao et al., 2020) CPU based machines require aggressive optimization techniques for both speed and network size. One popular speed optimization technique is NN quantization (Gholami et al., 2021; Kim et al., 2021; Zafrir et al., 2019), where network weights and activations are transformed from 32-bit floating-point representations to integers (typically 8-bit). Running inference using integer operations has two key advantages. First, the model size footprint is considerably reduced *e.g.* 8-bit quantization shrinks models by a factor of four. Second, inference throughput is significantly increased by using more efficient integer-based "single instruction multiple data" (SIMD) (Hennessy and Patterson, 2012) instructions while improving memory bandwidth utilization, which is typically a bottleneck limiting computational throughput for NNs (Quinn and Ballesteros, 2018).

Fundamentally, quantization leads to a quantitative loss of information due to the lowered numerical precision. As a result, applying integer quantization directly to NN models leads to considerable drop in accuracy (Zafrir et al., 2019). However, by carefully adjusting the quantization parameters such as the clipping thresholds, the accuracy loss can be significantly reduced, if not eliminated.

The majority of quantization research (Gholami et al., 2021) involve a mix of quantization-aware training (QAT) and post-training calibration techniques with varying complexities to resolve the quantization performance gap. Several works (Kim et al., 2021; Choi et al., 2018; Zhou et al., 2017; Choi et al., 2018; Krishnamoorthi, 2018; Louizos et al., 2019; McKinstry et al., 2019) detail techniques for QAT as well as approaches where the quantization parameters are optimized using statistics gathered during training. While these approaches typically close the gap in the quantized model accuracy, they requires access to the training pipeline as well as the training data. In addition, these methods are not applicable to black-box models where both training procedures and data are not available. Also, these methods may be affected by training instabilities, increasing the complexity of the training regimes as described in (Krishnamoorthi, 2018). Post-training approaches such as (Migacz, 2017; Bhandare et al., 2019) require calibration techniques on selected datasets. For example, in (Migacz, 2017) KL-divergence (Kullback and Leibler, 1951) between the unquantized and quantized activations on each layer was used to tune the quantization clipping thresholds. Special care needs to be taken when selecting a calibration dataset; as it needs to be diverse enough but yet task

[*] Equal contribution.

Proceedings of EMNLP 2022 Industry Track, pages 461–467
December 9–11, 2020. ©2022 Association for Computational Linguistics

specific. In certain cases this leads to low accuracy, or even unpredictable behaviour, if the run-time input deviates from the calibration dataset.

Two methods that share our high-level goals of eliminating the need for training datasets are introduced in (Nagel et al., 2019; Cai et al., 2020). These methods are implemented with CNN-based (Gehring et al., 2017) networks, and are used for image classification and object detection tasks. (Nagel et al., 2019) reduces the quantization error by rescaling the weights of consecutive CNN layers while taking advantage of the equivariance property of the piece-wise linear ReLU function. (Cai et al., 2020), on the other hand, tunes the quantization parameters using synthetic data generated utilizing mean and variance statistics obtained from the batch normalization layers of the model itself. While both methods are applicable for mainly CNN-based networks, our algorithm is considerably simpler to implement and targets transformers (Vaswani et al., 2017); particularly SOTA NLP networks with BERT-like (Devlin et al., 2018; Liu et al., 2019) pre-trained representations.

In this work, we present a method that utilizes the Interquartile Range (IQR) (Tukey et al., 1977; Rousseeuw and Croux, 1993), which is a measure of statistical dispersion, to clip the activations dynamically during inference time. Our method ensures that at least 75% of the token-wise extreme activations are not modified, while leaving the remaining 25% to be statistically modified as outliers, leading to a robust behaviour while considerably improving quantization accuracy. Our method works for any transformer-based "trained" model and does not require any form of training or calibration. Overall, our contributions can be summarized as follows:

- We propose a novel "ready-to-use" inference-time dynamic quantization method that does not require sophisticated re-training/fine-tuning and additional calibration strategies.
- Empirically our proposed model demonstrates both effectiveness and robustness on several different NLP benchmark tasks.
- Further, contrary to prior work, experiments suggest that our proposed method works both for monolingual and multilingual transformer architectures out-of-the-box.

2 Methodology

2.1 Backgound

Existing approaches to speeding up inference for Transformers mostly focus on General Matrix Multiply (GEMM) operations. Fast GEMM implementations routinely use GPU and CPU specific SIMD instructions, to execute many multiplications and additions in parallel. They also optimize memory access patterns to make the best use of available memory bandwidth. Integer quantization speeds up the GEMM operations by increasing the amount of data transferred with each memory transaction. They also take advantage of denser SIMD instructions. For example, 8-bit quantization packs four times the data per memory transaction compared to 32-bit floating point values. Many CPUs also support 8-bit SIMD multiplication operations, providing faster as well as cost-effective computation.

2.1.1 Uniform Quantization

Dynamic quantization for inference quantizes activations at run time. The model weights are typically quantized once ahead of execution. Let $\mathcal{M} \in \mathbb{R}^{m \times n}$ be a matrix of either an activation or parameter weights. The quantization scale (QS) is obtained as:

$$\mathrm{QS} = \max_{\substack{\forall i \in \{1,\ldots,m\} \\ \forall j \in \{1,\ldots,n\}}} |\mathcal{M}(i,j)|. \tag{1}$$

The matrix $\mathcal{M}$ is then quantized to $\bar{\mathcal{M}} \in \mathbb{Z}^{m \times n}$ as follows:

$$\bar{\mathcal{M}} = \mathrm{int}\left(\frac{2^b/2 - 1}{\mathrm{QS}}\mathcal{M}\right), \tag{2}$$

where b is the number of integerization bits, typically 8, and the function int is the element-wise integer conversion operator; e.g. a floor function. The reason for the subtraction by 1 in (2) is to ensure that the quantization range is equally spread around zero. In the case of 8 bits, the range becomes ± 127. This formulation also results in a symmetric form of uniform quantization, where the quantization is evenly split around zero. This can be modified by adding a zero-shift resulting in an asymmetric quantization (Krishnamoorthi, 2018), which may particularly be useful for certain activation functions such as ReLU (Nair and Hinton, 2010) and GELU (Hendrycks and Gimpel, 2016). While non-uniform quantization (Gholami et al., 2021) has been explored to better capture weight and activation distribution with variable step sizes, uniform quantization leads to more efficient implementation on current hardware such as GPUs and CPUs with acceptable accuracy. Once matrices are quantized, GEMM operations can be performed using integer arithmetic allowing the use of fast SIMD instruction sets.

Quantization lowers numerical precision which leads to loss of information. Examining (1) shows how the QS can increase precision errors if it takes extreme values that largely deviate from the majority activations. Therefore, the activation tensor must be clipped to reduce the quantization error; however, excessive clipping can lead to distortions in the activation which also leads to drops in accuracy.

In the following section, we will outline a method that chooses better QS values for each activation tensor dynamically during inference, without any modification to the training pipeline or any requirement for calibration procedures.

2.2 Interquartile Range Clipping

If we consider the extreme values in the activations as outliers in a distribution, there is a substantial amount of research for identifying outliers (Ben-Gal, 2005; Hodge and Austin, 2004). Our solution makes use of a low complexity univariate statistical-based method for outlier detection referred to as the Interquartile Range (IQR) method originally proposed by Tukey (Tukey et al., 1977).

IQR is also considered a robust statistical measure (Rousseeuw et al., 2011) of the data spread, with the notion of robustness being defined using the concept of a *breakdown point* (Rousseeuw and Croux, 1993; Rousseeuw et al., 2011). The breakdown point is the minimum number of data that can be arbitrarily replaced while keeping the statistical measure bounded. The sample mean and variance have a 0 breakdown point, meaning that these measures are changed by even a single outlier; on the other hand, the IQR has a 25% breakdown point, making it a stable measure even if up to 25% of the data are outliers.

We introduce an algorithm that effectively uses IQR to clip outliers from an activation tensor which consequently improves the selection of the quantization scale as in (1). It is worth noting that a direct implementation of the IQR method is too slow as it uses a sorting operation in order to identify the quartiles on the data. The complexity of a naive implementation would be $\mathcal{O}(N \log N)$ where N is the number of elements of the activation tensor. In the case of BERT-like models, $N = L \times H$, where L is the sequence length and H is the hidden dimension; *e.g.* for BERT-Large, $N = 512 \times 1024$. To lower this complexity, we obtain the IQR clipping threshold from a reduced set formed by taking the maximums, in absolute sense, along the H dimension. We will refer to this algorithm as the Token-Maximums IQR (TM-IQR) clipping. The resulting complexity of the IQR clipping becomes $\mathcal{O}(N + L \log L)$. Our experiments show that adding this form of IQR clipping slows inference by less than 2%, which is negligible considering the resulting accuracy gains.

Algorithm 2 Activation clipping using TM-IQR

```
    Input: Activation tensor $\mathcal{A} \in \mathbb{R}^{L \times H}$
    $\mathcal{L} \leftarrow \{1, 2, \ldots, L\}$
    $\mathcal{H} \leftarrow \{1, 2, \ldots, H\}$
1:  $M(i) \leftarrow \max_{\forall j \in \mathcal{H}} |\mathcal{A}(i,j)|, \forall i \in \mathcal{L}$
2:  $M \leftarrow \text{sort}(M)$
3:  $q1 \leftarrow \text{first-quartile}(M)$
4:  $q3 \leftarrow \text{third-quartile}(M)$
5:  $t \leftarrow q3 + 1.5(q3 - q1)$ -
6:  $\mathcal{A}(i,j) \leftarrow \min(\mathcal{A}(i,j), t), \quad \forall (i,j) \in \mathcal{L} \times \mathcal{H}$
7:  $\mathcal{A}(i,j) \leftarrow \max(\mathcal{A}(i,j), -t), \forall (i,j) \in \mathcal{L} \times \mathcal{H}$
    Return: $\mathcal{A}$
```

Algorithm 2 outlines the basic procedure of our TM-IQR clipping. In Line 1 we compose the set of token-maximum activations in the absolute sense. Essentially, we are reducing the set of activations to a smaller representative set that contains the top outliers of the larger set. Lines 2 to 5 compute the IQR threshold t which is then used to clip the entire activation tensor in lines 6 and 7. The value 1.5 in line 5 is commonly referred to as the IQR scale. It was historically proposed by Tukey (Tukey et al., 1977) as a level to detect outliers. It is possible to attempt to fine-tune this value, however we chose to use the historical value without tuning in line with the objective of our paper.

It is important to note that the TM-IQR algorithm assigns a dynamic clip value for each activation tensor as opposed to using a fixed value for all run-time inference. Unlike fixed clipping tuned by training datasets, we expect TM-IQR clipping to be applied in a zero-shot approach across multiple tasks while maintaining reasonable empirical accuracy. This is due to the fact that our clipping strategy guarantees that at least 75% of the row-wise extreme activations are not impacted by it, while a fixed clipping method does not offer such guarantees for all types of input, as is the

CPU	Precision	Method	Batch	WPS
Xeon 8260	int8	none	48	29005
Xeon 8260	int8	IQR	48	28640
V100	fp16	none	128	71998

Table 1: IQR throughput cost in WPS (words per sec) averaged over 4 runs. Each input is 512 tokens. 48 core Xeon 8260 and V100 speed included for reference.

case when the input is not very aligned with training data. This has the important effect of limiting the distortion error, which occurs when quantizing activations with excessive clipping.

3 Experimental Setup

Our run-time inference engine, implemented in C++, supports both FP32 and optimized 8-bit integer quantized inference (I8). We quantize model weights at load-time and dynamically quantize activations at run-time. The TM-IQR technique is a straightforward modification with a negligible impact on inference speed, as shown in Table 1.

3.1 TM-IQR

TM-IQR can be applied to the activations before each quantized GEMM operation. However, we found that the second feed-forward GEMM, henceforth referred to as FF2, contributes the majority of the quantization error. The input dimension of FF2 is very wide, $4 \times H$, providing more of a chance for saturation and integer numerical instability to accumulate. In addition, the input to FF2 constitutes the activations of either ReLU or a GELU non-linearities. The range of such activation functions is unbounded on the positive side, which further increases the chance of saturation. Therefore, we found it most effective to apply the TM-IQR to the input activations of the FF2 GEMM operation.

3.2 Tasks

We test our proposed method on GLUE (Wang et al., 2018) and 2 popular question answering (QA) tasks: Natural Questions (NQ) (Kwiatkowski et al., 2019) and TyDI [1] (Clark et al., 2020). We train all our tasks using the publicly available (Wolf et al., 2019). For GLUE tasks, we run 5 seeds with hyper-parameters using HuggingFace's defaults for BERT while tuning the learning rate for RoBERTa (refer to A for more details). For QA tasks, we follow (Alberti et al., 2019; Clark et al., 2020). Our underlying pre-trained language model for GLUE is both BERT (cased) (Devlin et al., 2018) and RoBERTa (Liu et al., 2019), while for QA, we used XLM-R (Conneau et al., 2019). Note our method *does not need* any fine-tuning once this step is done and models are obtained.

Task	FP32	I8	TM-IQR
XLM-R-base TyDI	67.7	62.9	**67.0**
XLM-R-large TyDI	68.8	66.8	**68.4**
XLM-R-base NQ	54.6	48.0	**53.4**
XLM-R-large NQ	56.6	53.3	**56.1**

Table 2: Question Answering performance.

4 Results

Since our method does not modify the training pipeline or tune the quantization parameters on training sets, we compare our results directly to the FP32 numbers. We are not expecting our method to outperform FP32 but rather to reduce the negative effect of quantization while keeping its speed as well as simplifying the model deployment process.

4.1 Question Answering

On TyDI and NQ (Table 2), TM-IQR clearly recovers most of the performance lost to dynamic quantization and is superior to I8 by 1 point on average. Similar to GLUE, TM-IQR still performs well with the I8 drop being the highest.

4.2 GLUE

Table 3 shows that TM-IQR is robust with an overall average score drop, compared to FP32, by *only* 0.2% for BERT-base, 0.5% for BERT-large, 1.2% for RoBERTa-base and 0.4% for RoBERTa-large. For all 4 pretrained models, TM-IQR wins on average. Even when TM-IQR does not outperform I8, the loss is relatively small. Interestingly, TM-IQR does well for cases where I8 drop is large, *e.g.* CoLA and RTE for all models and STS-B for RoBERTa-base.

5 Conclusion

We show that BERT-like models can be quantized to 8-bit integers with good accuracy without the need to modify training procedures or add extra data sets for parameter calibration. We present a robust statistically-based algorithm that dynamically adjusts the quantization clipping to maintain reasonable accuracy. Our empirical results demonstrate the effectiveness of our method on a number

[1] Note that TyDI is multilingual among 11 typologically diverse languages.

Task	FP32	I8	TM-IQR
BERT-base-cased			
MNLI	83.7 (0.2)	82.3 (0.5)	**83.5** (0.3)
MNLI-MM	84.1 (0.1)	82.9 (0.2)	**83.8** (0.2)
CoLA	58.0 (1.4)	48.3 (0.9)	**57.7** (1.6)
SST-2	92.3 (0.3)	**92.1** (0.2)	92.0 (0.4)
MRPC	88.5 (1.2)	**88.8** (1.6)	88.5 (1.5)
STS-B	88.3 (0.8)	87.7 (0.8)	**88.1** (0.8)
QQP	87.4 (0.1)	86.2 (0.3)	**87.2** (0.2)
QNLI	90.8 (0.2)	90.3 (0.1)	**90.5** (0.2)
RTE	64.6 (1.0)	63.9 (1.0)	**64.9** (1.6)
Average	82.0	80.3	**81.8**
BERT-large-cased			
MNLI	86.4 (0.1)	86.0 (0.2)	86.0 (0.1)
MNLI-MM	86.5 (0.2)	86.3 (0.1)	86.3 (0.2)
CoLA	62.9 (0.8)	60.6 (1.5)	**62.1** (1.2)
SST-2	93.3 (0.5)	92.8 (0.7)	**92.9** (0.4)
MRPC	90.5 (0.5)	89.6 (0.9)	**90.5** (0.7)
STS-B	89.6 (0.6)	87.4 (1.2)	**89.1** (0.3)
QQP	88.3 (0.2)	88.1 (0.1)	88.1 (0.1)
QNLI	92.4 (0.1)	91.9 (0.1)	**92.2** (0.2)
RTE	69.8 (1.4)	64.0 (2.0)	**68.5** (1.7)
Average	84.4	83.0	**84.0**
RoBERTa-base			
MNLI	87.0 (0.1)	85.8 (0.3)	**86.1** (0.1)
MNLI-MM	87.1 (0.1)	85.8 (0.2)	**86.1** (0.1)
CoLA	53.7 (1.9)	22.7 (4.7)	**50.8** (1.8)
SST-2	93.9 (0.2)	93.4 (0.4)	**93.5** (0.3)
MRPC	78.6 (2.7)	77.2 (1.1)	**78.4** (2.0)
STS-B	87.1 (0.8)	69.6 (0.8)	**85.5** (0.8)
QQP	88.3 (0.1)	87.2 (0.2)	**87.6** (0.1)
QNLI	92.5 (0.1)	90.1 (1.9)	**91.4** (0.3)
RTE	68.0 (2.1)	64.7 (2.5)	**67.4** (2.8)
Average	82.0	75.2	**80.8**
RoBERTa-large			
MNLI	90.6 (0.0)	90.0 (0.2)	**90.3** (0.1)
MNLI-MM	90.0 (0.3)	89.6 (0.1)	89.6 (0.2)
CoLA	63.5 (0.6)	63.1 (1.3)	**63.4** (0.6)
SST-2	96.3 (0.4)	95.8 (0.2)	95.8 (0.4)
MRPC	89.6 (0.4)	**90.1** (0.7)	88.7 (0.8)
STS-B	91.8 (0.1)	91.2 (0.2)	**91.4** (0.2)
QQP	89.8 (0.1)	89.4 (0.2)	89.4 (0.2)
QNLI	94.6 (0.2)	94.1 (0.3)	94.1 (0.3)
RTE	77.7 (2.0)	76.0 (5.1)	**77.3** (1.5)
Average	87.1	86.6	**86.7**

Table 3: The TM-IQR clipping algorithm on GLUE tasks with three computational modes, 32-bit floating-point (FP32), 8-bit quantization (I8) and our algorithm TM-IQR. Metric values are mean and standard deviation (in parenthesis) over 5 seeds.

of NLP monolingual and multilingual tasks, trained on both base and large size BERT-like models.

References

Chris Alberti, Kenton Lee, and Michael Collins. 2019. A bert baseline for the natural questions. *arXiv preprint arXiv:1901.08634*.

Irad Ben-Gal. 2005. Outlier detection. In *Data mining and knowledge discovery handbook*, pages 131–146. Springer.

Aishwarya Bhandare, Vamsi Sripathi, Deepthi Karkada, Vivek Menon, Sun Choi, Kushal Datta, and Vikram Saletore. 2019. Efficient 8-bit quantization of transformer neural machine language translation model. *CoRR*, abs/1906.00532.

Yaohui Cai, Zhewei Yao, Zhen Dong, Amir Gholami, Michael W Mahoney, and Kurt Keutzer. 2020. Zeroq: A novel zero shot quantization framework. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13169–13178.

Daniel Cer, Mona Diab, Eneko Agirre, Iñigo Lopez-Gazpio, and Lucia Specia. 2017. SemEval-2017 task 1: Semantic textual similarity multilingual and crosslingual focused evaluation. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 1–14, Vancouver, Canada. Association for Computational Linguistics.

Jungwook Choi, Zhuo Wang, Swagath Venkataramani, Pierce I-Jen Chuang, Vijayalakshmi Srinivasan, and Kailash Gopalakrishnan. 2018. PACT: parameterized clipping activation for quantized neural networks. *CoRR*, abs/1805.06085.

Jonathan H. Clark, Eunsol Choi, Michael Collins, Dan Garrette, Tom Kwiatkowski, Vitaly Nikolaev, and Jennimaria Palomaki. 2020. Tydi QA: A benchmark for information-seeking question answering in typologically diverse languages. *CoRR*, abs/2003.05002.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Unsupervised cross-lingual representation learning at scale. *arXiv preprint arXiv:1911.02116*.

Ido Dagan, Oren Glickman, and Bernardo Magnini. 2005. The pascal recognising textual entailment challenge. In *Machine Learning Challenges Workshop*, pages 177–190. Springer.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

William B. Dolan and Chris Brockett. 2005. Automatically constructing a corpus of sentential paraphrases. In *Proceedings of the Third International Workshop on Paraphrasing (IWP2005)*.

Jonas Gehring, Michael Auli, David Grangier, Denis Yarats, and Yann N Dauphin. 2017. Convolutional sequence to sequence learning. In *International Conference on Machine Learning*, pages 1243–1252. PMLR.

Amir Gholami, Sehoon Kim, Zhen Dong, Zhewei Yao, Michael W Mahoney, and Kurt Keutzer. 2021. A survey of quantization methods for efficient neural network inference. *arXiv preprint arXiv:2103.13630*.

Dan Hendrycks and Kevin Gimpel. 2016. Bridging nonlinearities and stochastic regularizers with gaussian error linear units. *CoRR*, abs/1606.08415.

John L. Hennessy and David A. Patterson. 2012. *Computer Architecture - A Quantitative Approach, 5th Edition*. Morgan Kaufmann.

Victoria Hodge and Jim Austin. 2004. A survey of outlier detection methodologies. *Artificial intelligence review*, 22(2):85–126.

S. Iyer, N.and Dandekar, and K. Csernai. 2017. First quora dataset release: Question pairs.

Xiaoqi Jiao, Yichun Yin, Lifeng Shang, Xin Jiang, Xiao Chen, Linlin Li, Fang Wang, and Qun Liu. 2020. Tinybert: Distilling bert for natural language understanding. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: Findings*, pages 4163–4174.

Sehoon Kim, Amir Gholami, Zhewei Yao, Michael W Mahoney, and Kurt Keutzer. 2021. I-bert: Integer-only bert quantization. *International Conference on Machine Learning*.

Raghuraman Krishnamoorthi. 2018. Quantizing deep convolutional networks for efficient inference: A whitepaper. *CoRR*, abs/1806.08342.

S. Kullback and R. A. Leibler. 1951. On information and sufficiency. *Annals of Mathematical Statistics*, 22(1):79–86.

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Matthew Kelcey, Jacob Devlin, Kenton Lee, Kristina N. Toutanova, Llion Jones, Ming-Wei Chang, Andrew Dai, Jakob Uszkoreit, Quoc Le, and Slav Petrov. 2019. Natural questions: a benchmark for question answering research. *Transactions of the Association of Computational Linguistics*.

Hector Levesque, Ernest Davis, and Leora Morgenstern. 2012. The winograd schema challenge. In *Thirteenth International Conference on the Principles of Knowledge Representation and Reasoning*.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Christos Louizos, Matthias Reisser, Tijmen Blankevoort, Efstratios Gavves, and Max Welling. 2019. Relaxed quantization for discretized neural networks. In *International Conference on Learning Representations*.

Jeffrey L McKinstry, Steven K Esser, Rathinakumar Appuswamy, Deepika Bablani, John V Arthur, Izzet B Yildiz, and Dharmendra S Modha. 2019. Discovering low-precision networks close to full-precision networks for efficient inference. In *2019 Fifth Workshop on Energy Efficient Machine Learning and Cognitive Computing-NeurIPS Edition (EMC2-NIPS)*, pages 6–9. IEEE.

Szymon Migacz. 2017. Nvidia 8-bit inference with TensorRT. In *GPU Technology Conference*.

Markus Nagel, Mart van Baalen, Tijmen Blankevoort, and Max Welling. 2019. Data-free quantization through weight equalization and bias correction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1325–1334.

Vinod Nair and Geoffrey E. Hinton. 2010. Rectified linear units improve restricted boltzmann machines. In *ICML*, pages 807–814. Omnipress.

Jerry Quinn and Miguel Ballesteros. 2018. Pieces of eight: 8-bit neural machine translation. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 3 (Industry Papers)*, pages 114–120, New Orleans - Louisiana. Association for Computational Linguistics.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. SQuAD: 100,000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2383–2392, Austin, Texas. Association for Computational Linguistics.

Sujith Ravi and Zornitsa Kozareva. 2021. SoDA: On-device conversational slot extraction. In *Proceedings of the 22nd Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 56–65.

Peter J Rousseeuw and Christophe Croux. 1993. Alternatives to the median absolute deviation. *Journal of the American Statistical association*, 88(424):1273–1283.

P.J. Rousseeuw, F.R. Hampel, E.M. Ronchetti, and W.A. Stahel. 2011. *Robust Statistics: The Approach Based on Influence Functions*. Wiley Series in Probability and Statistics. Wiley.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1631–1642, Seattle, Washington, USA. Association for Computational Linguistics.

John W Tukey et al. 1977. *Exploratory data analysis*, volume 2. Reading, Mass.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. 2018. GLUE: A multi-task benchmark and analysis platform for natural language understanding. *CoRR*, abs/1804.07461.

Alex Warstadt, Amanpreet Singh, and Samuel R. Bowman. 2019. Neural network acceptability judgments. *Transactions of the Association for Computational Linguistics*, 7:625–641.

Adina Williams, Nikita Nangia, and Samuel Bowman. 2018. A broad-coverage challenge corpus for sentence understanding through inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1112–1122, New Orleans, Louisiana. Association for Computational Linguistics.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, et al. 2019. Huggingface's transformers: State-of-the-art natural language processing. *arXiv preprint arXiv:1910.03771*.

Ofir Zafrir, Guy Boudoukh, Peter Izsak, and Moshe Wasserblat. 2019. Q8bert: Quantized 8bit bert. *arXiv preprint arXiv:1910.06188*.

Aojun Zhou, Anbang Yao, Yiwen Guo, Lin Xu, and Yurong Chen. 2017. Incremental network quantization: Towards lossless cnns with low-precision weights. *CoRR*, abs/1702.03044.

Jinhua Zhu, Yingce Xia, Lijun Wu, Di He, Tao Qin, Wengang Zhou, Houqiang Li, and Tieyan Liu. 2019. Incorporating bert into neural machine translation. In *International Conference on Learning Representations*.

A Evaluation on GLUE Task

For GLUE experiments we use the publicly available open-source library `PyTorch-Transformers` (Wolf et al., 2019). We report the standard metric on each task, specifically: Accuracy is used for MNLI, MNLI-MM (mismatch) (Williams et al., 2018), SST-2 (Socher et al., 2013), QNLI (Rajpurkar et al., 2016), and RTE (Dagan et al., 2005). Mathews correlation coefficient is used for CoLA (Warstadt et al., 2019). F1 is used for MRPC (Dolan and Brockett, 2005) and QQP (Iyer et al., 2017). Finally, Pearson correlation coefficient is used for STS-B (Cer et al., 2017), For BERT models, We use the default hyper-parameters provided by the HuggingFace's library, specifically the learning rate is $2. \times 10^{-5}$, the batch-size is 32 and the fine-tuning epochs is 3, except for MRPC where the the fine-tuning epochs is 5. For RoBERTa models, we tuned the learning rate in $[5e-7, 2e-6]$ for best devset results on FP32 evaluations, in addition we increase the epochs to 6 for the two large datasets, MNLI and QQP, and to 12 for the rest of the tasks. Similarly to (Kim et al., 2021) we exclude WNLI (Levesque et al., 2012) since it showed unstable results even on FP32 due to its small dataset.