ON QUANTIZATION OF IMAGE CLASSIFICATION NEURAL NETWORKS FOR COMPRESSION WITHOUT RETRAINING

Marcos Tonin and Ricardo L. de Queiroz

Electrical Engineering and Computer Science Department
Universidad de Brasilia, Brazil
marcosvtonin@gmail.com and queiroz@ieee.org

ABSTRACT

We studied the quantization of neural networks for their compression and representation without retraining. The goal is to facilitate neural network representation and deployment in standard formats so that general networks may have their weights quantized and entropy coded within the deployment format. We relate weight entropy and model accuracy and try to evaluate distribution of weights against known distributions. Many scalar quantization strategies were tested. We have found that weights are typically approximated by a Laplacian distribution for which optimal quantizers are approximated by entropy-coded uniform quantizers with dead-zones. Results indicate that it is possible to reduce 8-fold the size of the popular image classification networks with accuracy losses near 1%.

Index Terms—Neural network compression, weight quantization, ONNX file compression.

1. INTRODUCTION

The use of artificial intelligence (AI) to solve problems in diverse areas of knowledge is increasingly frequent and intense. One of the reasons for this high frequency is the applicability in different contexts and the ability to respond to real challenges [1], such as safe driving in autonomous cars [2].

AI often uses neural networks (NN) [1], which have been used in different applications, such as image classification, object detection, body analysis, machine comprehension, machine translation. NN models are made of weights, biases, computational units, and layers [1].

In Internet of Things (IoT) [3] and Edge Computing [4], typical equipments have limited resources and energy [5, 6]. Hence, the NN models should be compacted and reduced. These constraints have given rise to interest in NN compression. Since NN is used in the most diverse devices and varied occasions, the interoperability of the NN becomes a critical point in their development. The Open Neural Network eXchange (ONNX) format allows interoperability between frameworks to train the model in one tool and use another for inference and prediction [7]. ONNX works by defining a standard set of operators and default data types, in addition to weights and biases. The weights can be represented as 32-bit floating-point as specified in IEEE 754 [8], a float. There is also the possibility to store the parameters as a 64-bit floating-point, a double. There are ONNX models for object detection, gesture analysis, text translation, image classification, etc., for example, MobileNet, AlexNet, GoogLenet and VGG [7].

There are several works on NN compression. However, most involve retraining the model or changing its structure (e.g., insertion or removal of layers), such as the methods that use pruning [9, 10, 11, 12, 13], layer quantization [14], weight sharing [15] and quantization in general [16]. We are interested in compression without retraining. For example, Seo and Kim use a hybrid method with uniform compression followed by K-means clustering [17]. Dupuis et al. achieve compression by sharing weights among layers [18].

Among the compression with retraining efforts, we have the MPEG’s call for NN compression (MPEG–NNR), which aims at defining a compressed, interpretable and interoperable representation for trained NN [19]. Also, MPEG-NNR recommends interoperable formats like ONNX and NNEF (Neural Network Exchange Format) for a compressed representation of NN. MPEG-NNR’s call for proposal has the following requirements [19]: efficient representation of the model (the size of the compressed model has to be at least 30% smaller than the original model); to support different types of NN (CNN, RNN and others); the compressed representation may contain all the parameters and weights of the NN; the possibility of performing the inference of the compact model; the method to compress the NN independently of the dataset used to train the original model; low computational power and memory consumption to perform decoding.

We, however, are interested in reducing the size of networks in ONNX format without retraining them and without imposing a significant performance impact.

2. WEIGHT QUANTIZATION

If one is to look for data inter-dependence within NN, there must be an implicit ordering among the data. Weights and
biases of the NN model are separated into layers, which are usually ordered from the input to the output one. However, there is no exact ordering, and we could not find clear peaks in spectral analysis or correlation functions to warrant vector quantization or transformation. Therefore, as we found no dependence between the coefficients, we focus on scalar quantization.

**Table 1.** Some ONNX’s models.

<table>
<thead>
<tr>
<th>Models</th>
<th>File size (MB)</th>
<th>Percentage ratio of weights and biases in the file (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Caffenet</td>
<td>232.57</td>
<td>99.999</td>
</tr>
<tr>
<td>Efficientnet-lite4</td>
<td>49.54</td>
<td>99.841</td>
</tr>
<tr>
<td>GoogLenet</td>
<td>26.72</td>
<td>99.907</td>
</tr>
<tr>
<td>Mobilenet</td>
<td>13.59</td>
<td>99.366</td>
</tr>
<tr>
<td>Resnet</td>
<td>170.40</td>
<td>99.930</td>
</tr>
<tr>
<td>Shufflenet</td>
<td>5.46</td>
<td>99.246</td>
</tr>
<tr>
<td>SqueezeNet</td>
<td>4.73</td>
<td>99.713</td>
</tr>
<tr>
<td>VGG</td>
<td>548.15</td>
<td>99.997</td>
</tr>
</tbody>
</table>

Most of the data in an ONNX file are weights and biases, as shown in Table 1. Their amplitude distribution is usually concentrated around zero and is approximately symmetric around zero. The weights distribution of Googlenet and the Shufflenet models are shown in Fig. 1.

**Fig. 1.** Weights histogram for Googlenet and Shufflenet models.

We analyzed the weight histograms and compared them to known probability density functions (PDF), such as Alpha, Cauchy, Exponential, Logistic, Gamma, Laplace, Gaussian, and Uniform [20, 21, 22]. In order to evaluate distribution distances, we used two distance metrics: sum of squared errors (SSE) and Kullback-Leibler divergence ($KLD$). Let $P(i)$ be samples of the model distribution, while $Q(i)$ be similar samples of the reference, standard distribution. Then

$$SSE = \sum_{i=1}^{n} (P(i) - Q(i))^2,$$

$$KLD(P||Q) = \sum_{i=1}^{n} P(i) \log \frac{P(i)}{Q(i)}.$$  

Both $SSE$ and $KLD$ measure the similarity between two distributions. The $KLD$ measures the information lost when $Q$ is used to estimate $P$, and $SSE$ is the norm of the error in between distributions.

Tables 2 and 3 rank PDFs that are most similar to the weights histogram of NN models. In Table 2, referring to the SSE, we found that the Laplacian distribution is a good approximation since the first or second distribution best approximates the models. Moreover, the KLD metric (Table 3) has the Laplacian distribution always among the three best approximations to the models. Hence we can conclude that the Laplacian distribution is a reasonable approximation. Thus, we can use Sullivan’s results [23] for entropic coding of quantized values, which reports that, for Laplacian distributions, the optimal quantization is approximated by uniform quantization with dead-zones.

Uniform quantization can be midrange or midrise, as described in Table 4. With a dead-zone, the level around zero has a different range, as described in Table 4. In Table 4, $X$ represents the value of a weight to be quantized, $X_q$ the value passed to the decoder, and $\hat{X}$ the value reconstructed by the decoder. $\Delta$ is the size of the quantization steps and $s(X)$ is the sign function returning 1 if the number is positive and -1 otherwise. $\sigma$ defines the relative size of the dead-zone related to the quantization step. Note that $\sigma = 0$ implies the midrise quantizer, while the midrange quantizer has $\sigma = 0.5$. The $\lfloor X \rfloor$ operator is the ceiling (top) rounding of $X$.

In non-uniform quantization, step sizes are unequal, for...
example, using logarithmic functions. Usually, steps near the origin are smaller. We performed tests with the non-uniform "$\mu$-law" quantization. Another way to perform non-uniform quantization is to use different floating-point formats. The standard format is the 32-bit float [8]. There are IEEE-defined versions for 16-bits and 8-bits (half-precision and minifloat, respectively). Other float versions can be defined using the format:

$$s_0 e_0 e_1 \ldots e_{p-1} m_0 m_1 \ldots m_{q-1},$$

where the sign $s_0$ is a bit, followed by $p$ exponent bits and $q$ mantissa bits.

### 3. RESULTS

Tests were performed using the ImageNet Large Scale Visual Recognition Challenge 2012 (ILSVRC 2012) validation dataset, composed of 50000 images [24]. These images have a classification among 1000 possible classes. The networks used here are presented in Table 1 (all of them are CNN type in ONNX format). In order to evaluate the performance of the image classification models, we use the accuracy Top 1 in ONNX format). In order to evaluate the performance of the image classification models, we use the accuracy Top 1 and Top 5 classes. We have also computed the entropy of the images the model was able to correctly predict, considering otherwise. So, for the 50000 validation images, $acc_1$ indicates how many images the model was able to correctly predict, considering the top 5 classes. We have also computed the entropy of the NN weights, which estimates how many bits in average an encoder would spend to encode each quantized weight.

Table 4 shows entropy and accuracy results for the unquantized models. We evaluate the uniform quantization (midtread and midrise), non-uniform quantization (floats representations and "$\mu$-law") and, also dead-zone quantization ($\sigma$ as 0.1, 0.25, 0.4, and 0.7). Note that midrise is the case $\sigma = 0$ and midthread is the case $\sigma = 0.5$. For the midrise, midthread and dead-zone quantizations, we vary the number of bits from 24 to 2 bits ($2^b$, $2 \leq b \leq 24$), with the weights ranging from $-1$ to $+1$; As for the non-uniform ($\mu$-law) quantization, we vary $\mu$ from $2^{24}$ to $2^2$. The float representation was chosen with 16, 12, 10, 8, 7, 6, 5, 4 and 3 bits according to Eq. 3 and IEEE 754.

Figure 2 compares the $acc_1$ results for uniform quantization schemes. The RD results of the $acc_5$ metric has qualitatively similar behaviors to $acc_1$, but they are not quantitatively identical. Thus, when $acc_1$ tends to decrease, $acc_5$ also decreases at a slower rate. For simplicity, The results for the $acc_5$ metric were not shown. In Fig. 2, except for the Mobilenet case, a better result is achieved for midrise, midthread, or dead-zone with $\sigma = 0.4$. For most networks, the results do not vary much. The Mobilenet network is the most sensitive to dead-zone step size. For this model, $\sigma = 0.7$ yields the best results.

Table 5 refers to the point with the best performance shown in Fig. 2. The best performance point is the point with the lowest entropy, such that there is a maximum difference of 1% in the $acc_1$ and $acc_5$ metrics compared to the model’s initial accuracy.

Figure 2 and Table 6, indicate that a better result is achieved for midrise, midthread, or dead-zone with $\sigma = 0.4$ or $\sigma = 0.7$. For most networks, the results do not vary much.

### Table 4. Uniform Quantization Formulas.

<table>
<thead>
<tr>
<th>Method</th>
<th>$X_q$</th>
<th>$\hat{X}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Midtread</td>
<td>$\text{round}(\frac{X}{2})$</td>
<td>$\Delta X_q$</td>
</tr>
<tr>
<td>Midrise</td>
<td>$s(X)\lceil \frac{X}{\Delta} \rceil$</td>
<td>$s(X_q)\Delta(X_q - \frac{1}{2})$</td>
</tr>
<tr>
<td>Dead-zone</td>
<td>$\begin{cases} 0,</td>
<td>X</td>
</tr>
</tbody>
</table>

### Table 5. Entropy and accuracy results for uncompressed models.

<table>
<thead>
<tr>
<th>Models (Method)</th>
<th>Level Bits</th>
<th>Entropy achieved</th>
<th>$acc_5$ (%) achieved</th>
<th>$acc_1$ (%) achieved</th>
</tr>
</thead>
<tbody>
<tr>
<td>caffenet (Midtread)</td>
<td>8</td>
<td>1.828</td>
<td>78.958</td>
<td>55.5</td>
</tr>
<tr>
<td>efficientnet-lite ($\sigma = 0.7$)</td>
<td>9</td>
<td>6.528</td>
<td>93.374</td>
<td>77.194</td>
</tr>
<tr>
<td>googlenet ($\sigma = 0.4$)</td>
<td>8</td>
<td>3.06</td>
<td>87.938</td>
<td>67.03</td>
</tr>
<tr>
<td>MobileNet ($\sigma = 0.7$)</td>
<td>14</td>
<td>11.914</td>
<td>88.482</td>
<td>68.304</td>
</tr>
<tr>
<td>resnet ($\sigma = 0.7$)</td>
<td>13</td>
<td>7.736</td>
<td>93.322</td>
<td>76.632</td>
</tr>
<tr>
<td>shufflenet (Midtread)</td>
<td>9</td>
<td>6.078</td>
<td>67.512</td>
<td>41.77</td>
</tr>
<tr>
<td>squeezenet (Midtread)</td>
<td>8</td>
<td>4.67</td>
<td>76.578</td>
<td>53.2</td>
</tr>
<tr>
<td>VGG ($\sigma = 0.7$)</td>
<td>9</td>
<td>2.653</td>
<td>91.698</td>
<td>73.576</td>
</tr>
</tbody>
</table>

Table 6 refers to the point with the best performance shown in Fig. 2. The best performance point is the point with the lowest entropy, such that there is a maximum difference of 1% in the $acc_1$ and $acc_5$ metrics compared to the model’s initial accuracy.
Fig. 2. Rate and distortion (RD) comparisons (entropy $\times$ accuracy) among uniform and non-uniform quantization for different networks. Results for the $acc_1$ metric are shown, since graphics for the $acc_5$ are very similar.

(considering midrise, midthread and dead-zone methods). The network that shows the most divergence of results is Mobilenet, which has a noticeable difference in behaviour, when comparing dead-zone methods against non-uniform ones, midrise and midthread. In this case, the dead-zone quantizer with $\sigma = 0.7$ yields the best performance.

4. CONCLUSIONS

Results in Table 6 and Fig. 2 show us that, for most networks, it is possible to get rates close to 5 bits by weight without causing significant losses. For $acc_1$, as for $acc_5$, a maximum accuracy of 1.0% was achieved. After choosing the best quantizer and step size, an average of 5.62 bits/weight was achieved, representing a $5.6 \times$ reduction in the size of the NN originally with 32 bits. The proposed method can be useful to achieve the objectives proposed in the MPEG-NNR call [19]. For simplicity, only 8 NN models have been shown. We have tested others models, including non-image-related ones, with similar results. We hope these results could be characteristic and provide a general trend.

Future work may include tests with more NNs, and combine quantization with retraining methods.

5. REFERENCES


