Compression of Acoustic Event Detection Models with Low-rank Matrix Factorization and Quantization Training

Bowen Shi  
Toyota Technological Institute At Chicago  
bshi@ttic.edu

Ming Sun  
Amazon  
mingsun@amazon.com

Chieh-Chi Kao  
Amazon  
chiehchi@amazon.com

Viktor Rozgic  
Amazon  
rozgicv@amazon.com

Spyros Matsoukas  
Amazon  
matsouka@amazon.com

Chao Wang  
Amazon  
wngcha@amazon.com

Abstract

In this paper, we present a compression approach based on the combination of low-rank matrix factorization and quantization training, to reduce complexity for neural network based acoustic event detection (AED) models. Our experimental results show this combined compression approach is very effective. For a three-layer long short-term memory (LSTM) based AED model, the original model size can be reduced to 1% with negligible loss of accuracy. Our approach enables the feasibility of deploying AED for resource-constraint applications.

1 Introduction

Acoustic event detection (AED), the task of detecting whether certain events occur in an audio clip, can be applied in many industry applications [1,2,3,23]. The accuracy of AED models have been increased in a large scale in recent years based on deep learning approaches. However, to ensure high performance, those models are of large scale computation and memory intense, which makes it a challenge to deploy for real industrial applications with limited computation resources and memory. Our paper is focused on increasing the computation efficiency for AED models while maintaining their accuracies, so that AED deployment for resource-constraint industrial applications is feasible.

Compression of neural networks has been explored in broad context. We focus on two widely used and effective methods for deep models: low-rank matrix factorization and and quantization. Singular value decomposition (SVD) is a common factorization technique and has been explored in feedforward networks [9,10,21,22] and recurrent neural networks (RNN) [11]. Neural network quantization refers to compressing the original network by reducing number of bits required to represent weight matrices, and it has been studied for different model architectures [12,13,14,15,16,19,20]. By reducing the bit-width of weights, model size is reduced, and it also brings considerable acceleration via efficient low bit-width arithmetic operations supports available on hardware. For the quantization approach, It is important to fine-tune models with quantized weights to reduce the performance loss with quantized networks. Here we refer quantization with fine-turning as quantization training.
2 Methods

We start by formulating the multi-class audio event detection problem. Given an audio signal \( I \) (e.g. log mel-filter bank energies (LFBES)), the task is to train a model \( f \) to predict a multi-hot vector \( y \in \{0, 1\}^C \), with \( C \) being the size of event set \( \mathcal{E} \) and \( y_c \) being a binary indicator whether event \( c \) is present in \( I \). We denote \( \mathcal{D}_L = \{(I, y)\} \) as the labeled training dataset. Model \( f \) is trained using cross-entropy loss: \( L = - \sum_{(I, y) \in \mathcal{D}_L} \sum_{c=1}^C w_c y_c \log f_c(I) + (1 - y_c) \log(1 - f_c(I)) \), where \( w_c \) is the penalty of positive mis-classification of class \( c \). Specifically we focus on the RNN-based model in this paper. Compared to CNNs, it is more memory efficient and easier to deploy on devices with constrained resources.

Low-rank matrix factorization The factorization of weight matrices is based on the SVD compression of LSTM [11]. Let \( W^l \) and \( W^r \) denote the input and recurrent matrix of layer \( l \), respectively. We first run SVD on \( W^l \) to retain the top-\( r \) singular values and singular vectors, and its top-\( r \) right singular vectors \( V^l \) are shared with input weight matrix (see equation 1).

\[
\begin{align*}
W^l_h &= U^l_h \Sigma^l_h V^l_h^T, \\
W^r_z &= U^r_z \Sigma^r_z V^r_z^T, \\
\hat{y} &= e^{W^l_h x + W^r_z h}, \quad \text{where} \quad \hat{y}^l = \hat{y}^r
\end{align*}
\]

Quantization training Quantization refers to representing floating-point values with \( n \)-bit integers (\( n < 32 \)), as formulated in [2]. Note the scaling factor \( \alpha \) and minimum value \( \beta \) in equation [2] are not quantized.

\[
\hat{V} = \frac{V - \beta}{\alpha}, \quad \hat{Q}_n(\hat{V}) = \frac{\hat{V} (2^n - 1)}{2^n - 1}, \quad Q_n(V) = \alpha \hat{Q}_n(\hat{V}) + \beta
\]

As the quantization function \( \hat{Q}_n \) is discrete, its gradient is almost zero everywhere. To solve this problem, we apply straight-through estimator [18] to approximate the gradient of full-precision parameter \( \frac{\partial \hat{V}}{\partial V} \) with gradient of quantized value \( \frac{\partial V}{\partial \hat{V}} \) in the fine-tuning.

To combine low-rank matrix factorization and quantization training, the quantization operator \( \hat{Q}_n \) is applied to \( \hat{y}^l \), \( \hat{y}^r \) and \( V^l \). We quantize both model parameters and inputs. The original RNN is first trained in full-precision until convergence. After the low-rank matrix factorization is applied (equation [1]), the model is quantized and fine-tuned with quantization training. We find the fine-tuning step is important to maintain performance with quantization.

3 Experiments

The dataset used in our experiments is a subset of Audioset [17], which includes a large collection of 10-second audio segments for 632 categories of events. In particular, we select sounds of dog, baby crying and gunshots as our target events. They include both human and non-human vocals, as well as different sound event durations. The three events included in Audioset amount to 13,460, 2,313 and 4,083 audio segments, respectively. We also randomly select 36,036 examples from all other audio clips in Audioset as negative samples. The whole selected dataset is randomly split for training (70%), validation (10%) and test (20%) for each target events. 64 dimensional LFBE features are extracted for each audio clip, with window size of 25 ms and hop size of 10 ms, which are further normalized by global cepstral mean and variance normalization (CMVN).

Our baseline model is a three-layer LSTM with 256 hidden units in each layer. Dropout is added between layers at rate of 0.2. Adam optimizer is used with learning rate fixed at 0.001. The evaluation is based on detection error tradeoff (DET) curve (false negative rate vs. false positive rate). We compute area under curve (AUC) and equal error rate (EER) as the two quantitative measures.

As the distribution of weight matrices’ eigenvalues can be different across different LSTM layers, we follow the practice of [11] to set the same threshold \( \tau \) across layers as the fraction of retained singular values, defined as

\[
\tau = \frac{1}{N} \sum_{j=1}^N \sigma_j^2 \left/ \sum_{j=1}^N \sigma_j^2 \right., \quad \sigma_1 \geq \sigma_2 \geq \ldots \geq \sigma_N.
\]
Table 1 summarizes the results of low-rank matrix factorization compared to our baseline 3-layer LSTM. There is no accuracy degradation when $\tau$ is reduced to 0.6, which we hypothesize to be related to the regularizing effects. Table 2 summarizes the results with quantization compared to our baseline. Post-mortem (PM) refers to the case that quantization is only applied during inference, while quantization training (QT) refers to the case model fine-tuning is further performed on quantized weights. Our quantization training approach outperforms PM significantly for the 4-bit quantization case. We also note the simple PM quantization preserves the accuracy well (3.0% drop in AUC and 2.7% drop in EER) with 8-bit quantization.

Table 1: AUCs and EERs on test set with different $\tau$ on full-precision low-rank matrix factorization of three-layer LSTM. Lower AUC and EER indicate better performance.

<table>
<thead>
<tr>
<th>$\tau$</th>
<th>Params (MB)</th>
<th>Dog</th>
<th>Baby</th>
<th>Gunshot</th>
<th>Avg</th>
<th>Dog</th>
<th>Baby</th>
<th>Gunshot</th>
<th>Avg</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.0</td>
<td>5.273</td>
<td>7.49</td>
<td>6.34</td>
<td>5.10</td>
<td>6.31</td>
<td>15.19</td>
<td>13.58</td>
<td>11.79</td>
<td>13.52</td>
</tr>
<tr>
<td>0.9</td>
<td>2.396</td>
<td>7.22</td>
<td>7.08</td>
<td>4.78</td>
<td>6.36</td>
<td>14.54</td>
<td>15.23</td>
<td>10.96</td>
<td>13.54</td>
</tr>
<tr>
<td>0.8</td>
<td>1.069</td>
<td>7.61</td>
<td>6.36</td>
<td>4.88</td>
<td>6.28</td>
<td>15.26</td>
<td>13.52</td>
<td>11.05</td>
<td>13.27</td>
</tr>
<tr>
<td>0.6</td>
<td>0.316</td>
<td>7.74</td>
<td>6.71</td>
<td>5.07</td>
<td>6.51</td>
<td>15.54</td>
<td>13.8</td>
<td>11.29</td>
<td>13.54</td>
</tr>
<tr>
<td>0.4</td>
<td>0.132</td>
<td>10.79</td>
<td>10.58</td>
<td>10.87</td>
<td>10.75</td>
<td>19.39</td>
<td>18.72</td>
<td>18.74</td>
<td>18.95</td>
</tr>
</tbody>
</table>

Table 2: AUCs and EERs on test set w/ different quantization bits on three-layer LSTM without low-rank factorization. Lower AUC and EER indicate better performance.

<table>
<thead>
<tr>
<th># bits, type</th>
<th>Params (MB)</th>
<th>Dog</th>
<th>Baby</th>
<th>Gunshot</th>
<th>Avg</th>
<th>Dog</th>
<th>Baby</th>
<th>Gunshot</th>
<th>Avg</th>
</tr>
</thead>
<tbody>
<tr>
<td>Full-precision</td>
<td>5.273</td>
<td>7.49</td>
<td>6.34</td>
<td>5.10</td>
<td>6.31</td>
<td>15.19</td>
<td>13.58</td>
<td>11.79</td>
<td>13.52</td>
</tr>
<tr>
<td>8-bit PM</td>
<td>1.318</td>
<td>7.73</td>
<td>6.52</td>
<td>5.18</td>
<td>6.48</td>
<td>15.43</td>
<td>13.64</td>
<td>12.39</td>
<td>13.89</td>
</tr>
<tr>
<td>4-bit QT</td>
<td>0.659</td>
<td>8.15</td>
<td>8.08</td>
<td>6.05</td>
<td>7.43</td>
<td>15.91</td>
<td>15.45</td>
<td>13.08</td>
<td>14.81</td>
</tr>
</tbody>
</table>

Finally, we combine both low-rank matrix factorization and quantization training. Its results are summarized in table 3. The attained singular value ratio is fixed to be 0.6, as parameter size and accuracy is well balanced at this point according to table 1. When the model is quantized to 8-bit QT ($\approx 1.5\%$ of original size), AUC is only increased by $0.2\%$ and EER is even improved by a small margin ($1.9\%$). Performance is significantly degraded for 8-bit PM, which is related to the relatively highly compactness and unbounded intermediate outputs in the low-rank setting. Fine-tuning the quantized model is an essential step to reduce this effect. Though the performance is decreased for the 4-bit QT (less than $1\%$ of original model size), its relative degradation is much smaller compared to the 4-bit PM.

Table 3: AUCs and EERs on test set with different quantization bits on low-rank factorized three-layer LSTM, $\tau = 0.6$. Lower AUC and EER indicate better performance.

<table>
<thead>
<tr>
<th># bits, type</th>
<th>Params (MB)</th>
<th>Dog</th>
<th>Baby</th>
<th>Gunshot</th>
<th>Avg</th>
<th>Dog</th>
<th>Baby</th>
<th>Gunshot</th>
<th>Avg</th>
</tr>
</thead>
<tbody>
<tr>
<td>Baseline</td>
<td>5.273</td>
<td>7.49</td>
<td>6.34</td>
<td>5.10</td>
<td>6.31</td>
<td>15.19</td>
<td>13.58</td>
<td>11.79</td>
<td>13.52</td>
</tr>
<tr>
<td>8-bit PM</td>
<td>0.079</td>
<td>9.74</td>
<td>8.28</td>
<td>6.61</td>
<td>8.21</td>
<td>18.18</td>
<td>15.55</td>
<td>12.89</td>
<td>15.54</td>
</tr>
<tr>
<td>8-bit QT</td>
<td>0.079</td>
<td>7.37</td>
<td>6.38</td>
<td>5.24</td>
<td>6.33</td>
<td>15.11</td>
<td>13.34</td>
<td>11.31</td>
<td>13.26</td>
</tr>
<tr>
<td>4-bit PM</td>
<td>0.040</td>
<td>15.26</td>
<td>20.89</td>
<td>12.98</td>
<td>16.38</td>
<td>23.55</td>
<td>26.38</td>
<td>20.53</td>
<td>23.48</td>
</tr>
<tr>
<td>4-bit QT</td>
<td>0.040</td>
<td>9.10</td>
<td>8.13</td>
<td>7.29</td>
<td>8.17</td>
<td>17.05</td>
<td>14.82</td>
<td>12.97</td>
<td>14.94</td>
</tr>
</tbody>
</table>

4 Conclusion

In this paper we present a simple yet effective compression technique combining low-rank matrix factorization and quantization training. The proposed technique is applied to a multi-layer LSTM for AED. It compresses the AED model size to less than $1\%$ of the original, with AUC and EER performance well maintained. Model fine-tuning with quantization (QT) outperforms a naive quantization scheme (PM), which shows its consistent advantage with different number of quantization bits.
References


