Distributed Neural Network with TensorFlow on Human Activity Recognition over Multicore TPU

Dec. 20-23, 2021

Haklin Kimm, Incheon Paik

Department of Computer Science, East Stroudsburg University
The University of Aizu
Fukushima, Japan
OUTLINE

➤ Introduction
➤ Related Work
➤ Methods
➤ Experiment
➤ Result and Evaluation
➤ Conclusion & Future Work
INTRODUCTION
Introduction

- Increasing interests and success of applying deep learning neural networks to their big data platforms and workflows
- Distributed Long Short-Term Memory (dLSTM) neural network model using TensorFlow over multicore Tensor Processing Unit (TPU) on Google Cloud
- Application To Human Activity Recognition
Related Work

- **HAR research Using DNN:**
  - 3D CNN was first introduced by Tran and others [17].
  - A recurrent 3D CNN is applied for detecting hand gestures by Molchanov et al. [18].
  - LSTM (long short-term memory) is also applied to process the sequence information as shown in [18].
  - EmbraceNet and DenseNet have been proposed for the task with the CNN [2]. It is also shown in [19-20] that BERT or CNN along with LSTM works well learning sequence of languages, images, or signals.

- **Distributed Deep Learning:**
  - Data-parallel training and Scheduler issue [12-13]
Distributed Deep Learning with Tensorflow
## UCI-HAR Dataset

<table>
<thead>
<tr>
<th>Activities</th>
<th>Samples</th>
<th>Percentage</th>
</tr>
</thead>
<tbody>
<tr>
<td>Stand</td>
<td>138,105</td>
<td>18.5%</td>
</tr>
<tr>
<td>Lay</td>
<td>136,865</td>
<td>18.3%</td>
</tr>
<tr>
<td>Walk</td>
<td>122,091</td>
<td>16.3%</td>
</tr>
<tr>
<td>Down</td>
<td>107,961</td>
<td>14.4%</td>
</tr>
<tr>
<td>Up</td>
<td>116,707</td>
<td>15.6%</td>
</tr>
<tr>
<td>Sit</td>
<td>126,677</td>
<td>16.9%</td>
</tr>
</tbody>
</table>
METHOD
## DL Architecture

- Using Bidirectional LSTM

---

### Parameters of Each Layer of the Distributed DL

<table>
<thead>
<tr>
<th>Layer (type)</th>
<th>Output Shape</th>
<th>Param #</th>
</tr>
</thead>
<tbody>
<tr>
<td>bidirectional</td>
<td>(Bidirectional (None, 128, 256))</td>
<td>141312</td>
</tr>
<tr>
<td>bidirectional_1</td>
<td>(Bidirectional (None, 128, 256))</td>
<td>394240</td>
</tr>
<tr>
<td>lstm_2 (LSTM)</td>
<td>(None, 64)</td>
<td>82176</td>
</tr>
<tr>
<td>dense (Dense)</td>
<td>(None, 6)</td>
<td>390</td>
</tr>
</tbody>
</table>

Total params: 618,118  
Trainable params: 618,118  
Non-trainable params: 0
Distributed DL Implementation on Colab

- Distributed DL Model with TPU
- Distributed DL Model with CPU – On Google Colab
EXPERIMENT & EVALUATION
Evaluation Metrics for Distributed DL-TPU
(a) **Loss over epochs**

(b) **Accuracy over epochs**

(c) **Precision over epochs**

(d) **Recall over epochs**

**Evaluation Metrics for DL-CPU**
Performance Comparison

- DL-TPU’s total run time of 30 epochs is 203.868 seconds with average accuracy of 89.922, precision 90.221, recall 89.854; its F1 score is 90.037.
- DL-CPU’s total run time of 30 epochs is 5158.278 seconds with average accuracy of 92.399, precision 92.488, recall 92.331; its F1 score is 92.331.

<table>
<thead>
<tr>
<th>Model</th>
<th>Accuracy</th>
<th>Precision</th>
<th>Recall</th>
<th>F1 Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>TPU-DDL</td>
<td>89.92</td>
<td>90.22</td>
<td>89.85</td>
<td>90.04</td>
</tr>
<tr>
<td>CPU-DL</td>
<td>92.40</td>
<td>92.49</td>
<td>92.33</td>
<td>92.33</td>
</tr>
</tbody>
</table>
CONCLUSION
Conclusion & Future Work

- DDL (Distributed Deep Learning) model, built with bidirectional LSTM layer model, using TensorFlow which has been applied to a Google TPU (Tensor Processing Unit) equipped with 8 cores.

- UCI-HAR dataset used. (training set 71% and test set 29%; the total number of samples in this dataset is 748406)

- DDL-TPU shows the elapsed time of 203.868 seconds over 30 epochs, and DL-CPU provides the elapsed time of 5158.278 seconds. When calculating the speed-up ratio between the two models, it is about 25 times faster with about 2.5% F1 score decrease.

- Future Work:
  - Improvement in BERT Architecture and Distributed model.
THANKS!