IEEE 14th International Symposium on Embedded Multicore/Many-core Systems-on-Chip (MCSoC-2021)
Singapore University of Technology and Design, Singapore
December 20-23, 2021

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

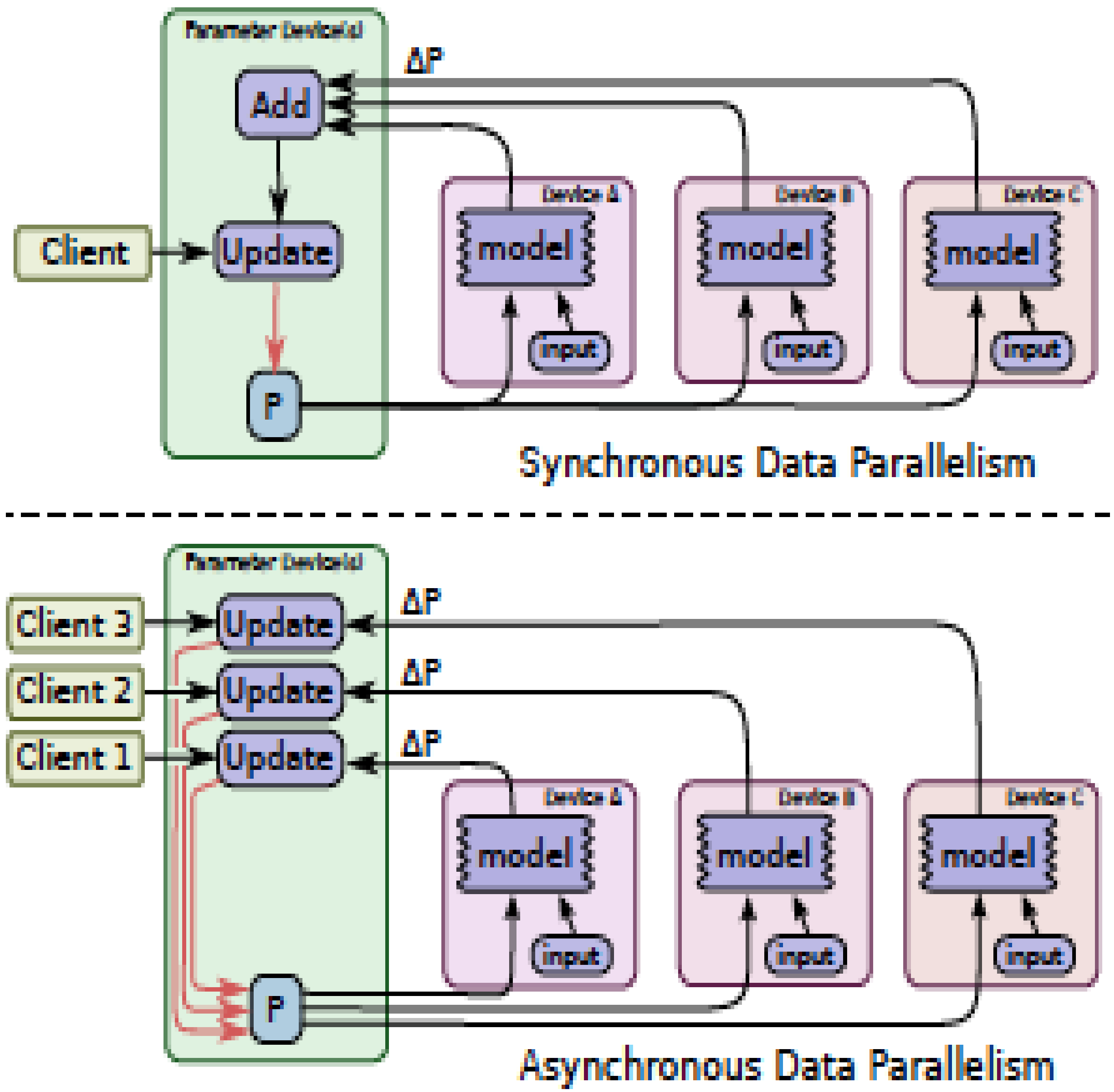
Introduction

- 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

Activities	Samples	Percentage
Stand	138,105	18.5%
Lay	136,865	18.3%
Walk	122,091	16.3%
Down	107,961	14.4%
Up	116,707	15.6%
Sit	126,677	16.9%

METHOD

DL Architecture

- Using Bidirectional LSTM

Parameters of Each Layer of the Distributed DL

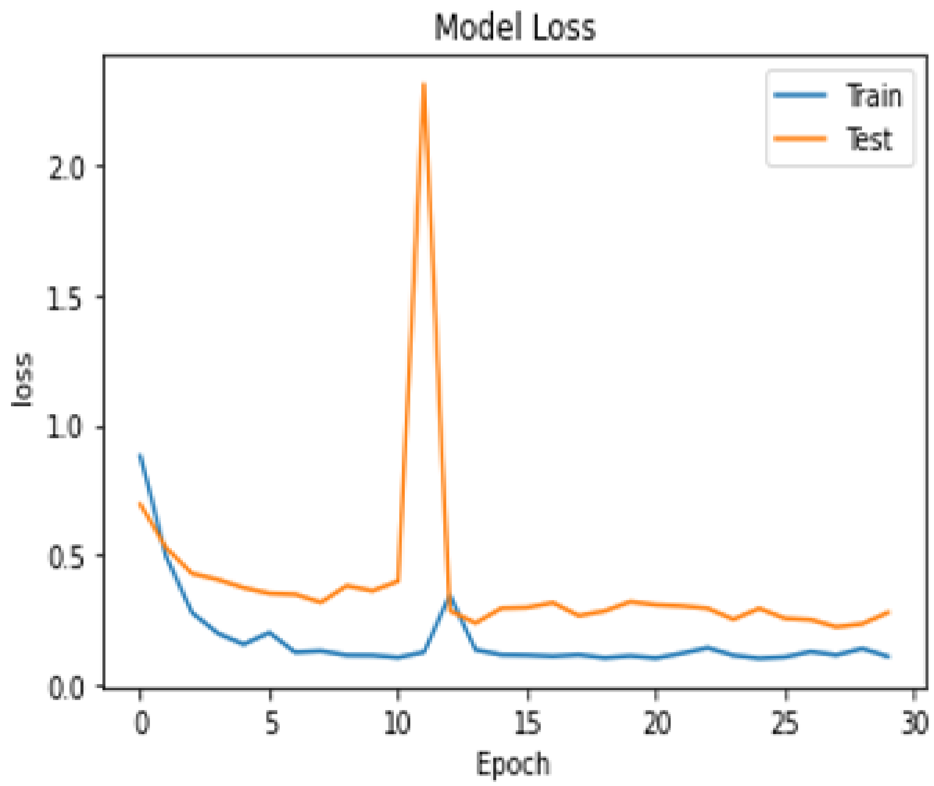
Layer (type)	Output Shape	Param #
bidirectional	(Bidirectional (None, 128, 256)	141312
bidirectional_1	(Bidirectional (None, 128, 256)	394240
lstm_2 (LSTM)	(None, 64)	82176
dense (Dense)	(None, 6)	390

Total params: 618,118
Trainable params: 618,118
Non-trainable params: 0

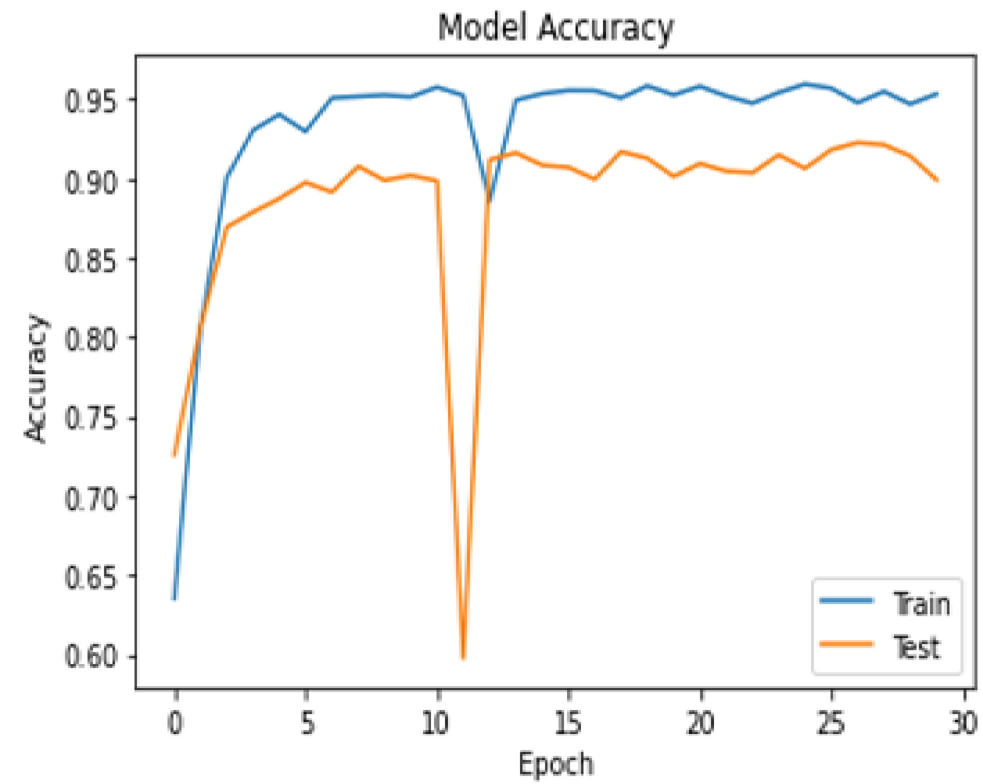
Distributed DL Implementation on Colab

- Distributed DL Model with TPU
- Distributed DL Model with CPU – On Google Colab

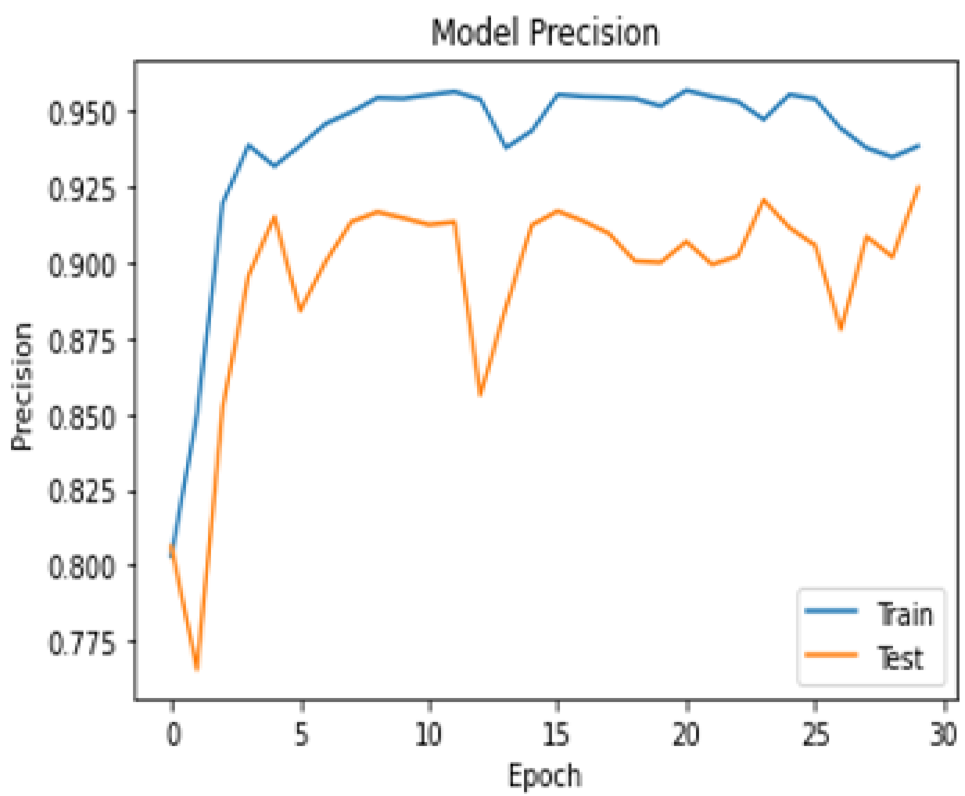
EXPERIMENT & EVALUATION



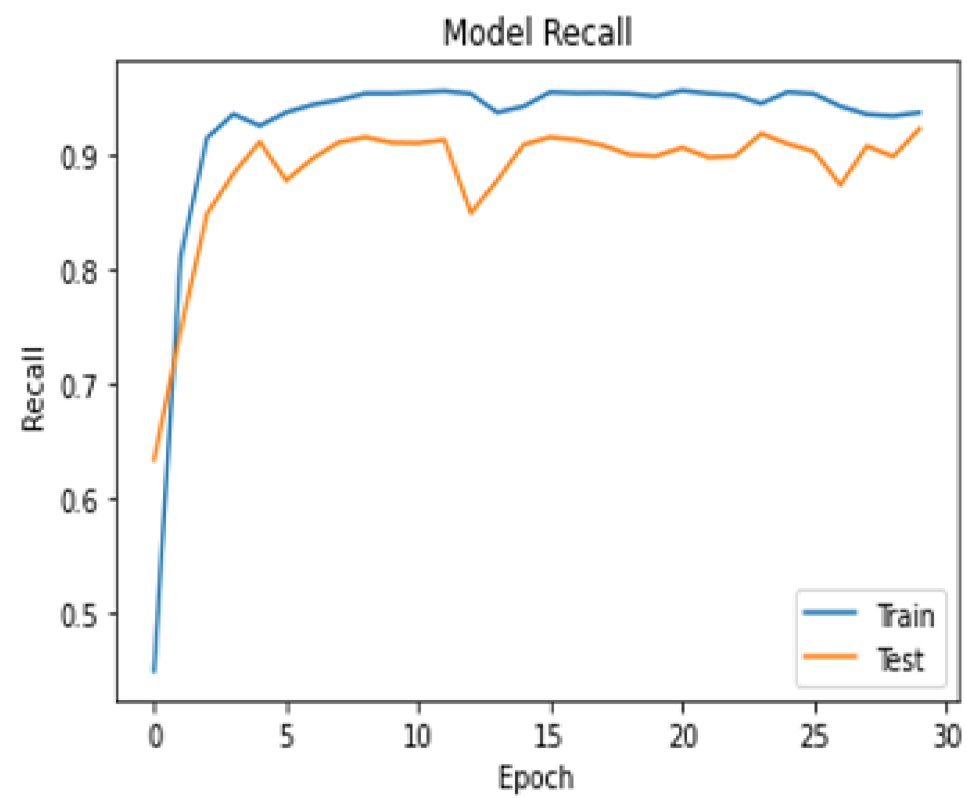
(a) Loss over epochs



(b) Accuracy over epochs

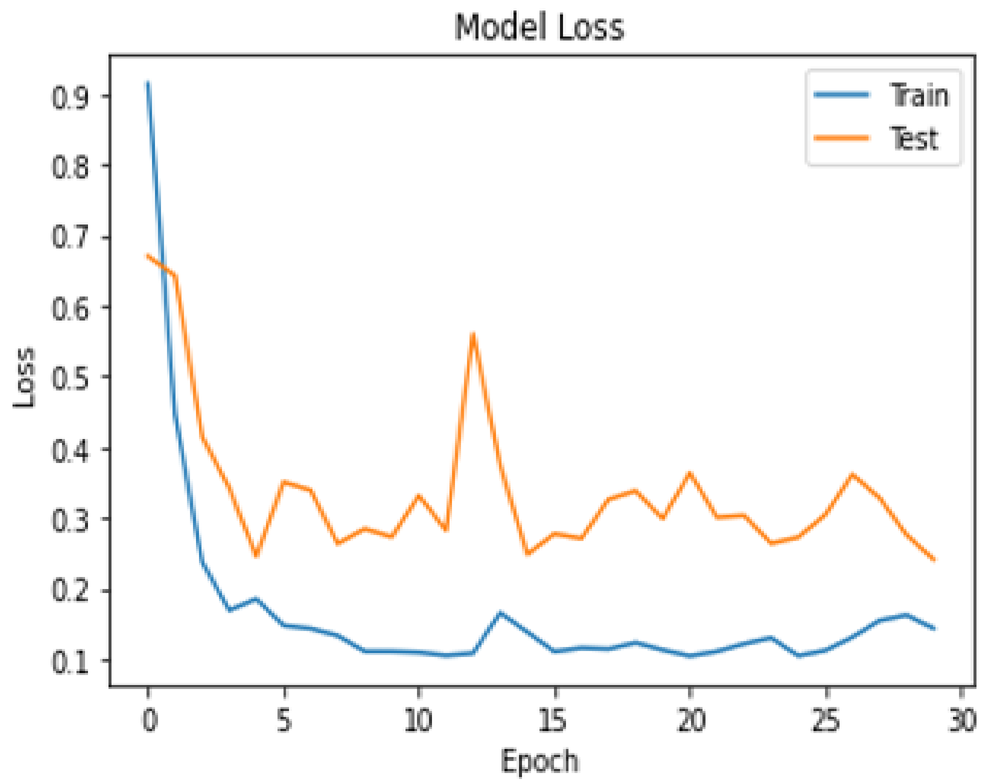


(c) Precision over epochs

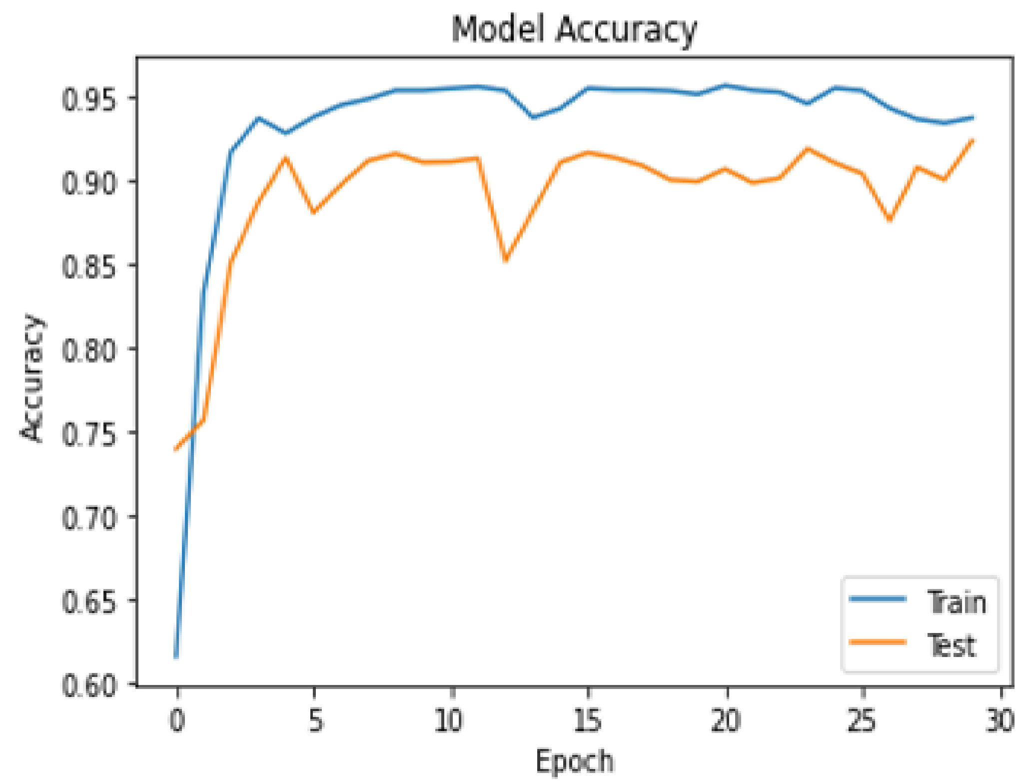


(d) Recall over epochs

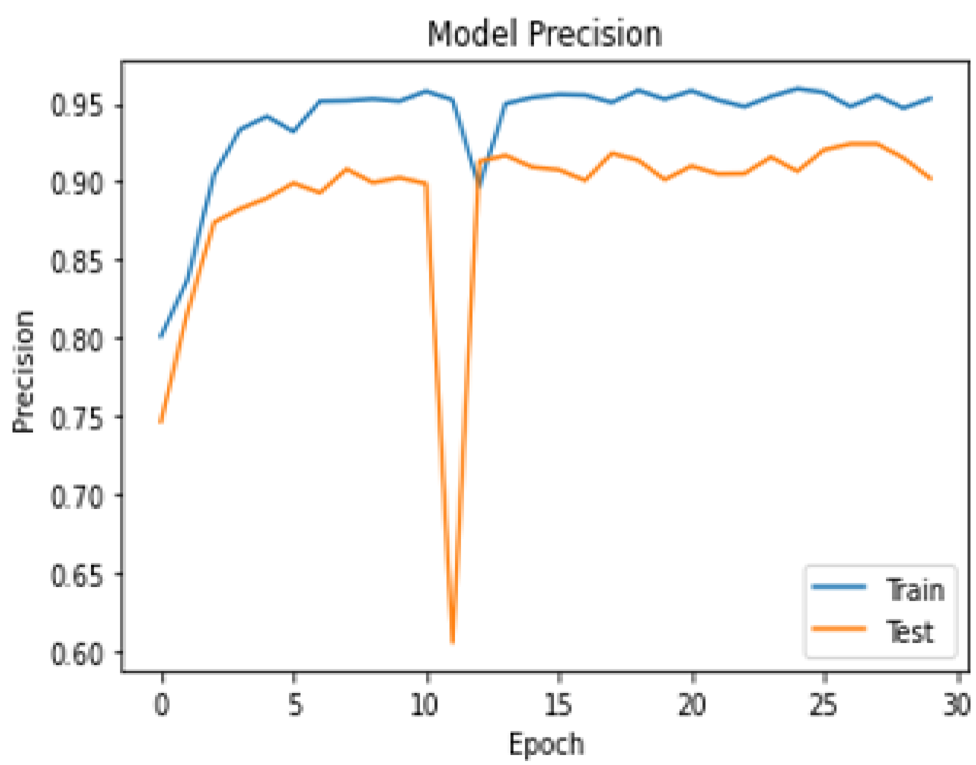
Evaluation Metrics for Distributed DL-TPU



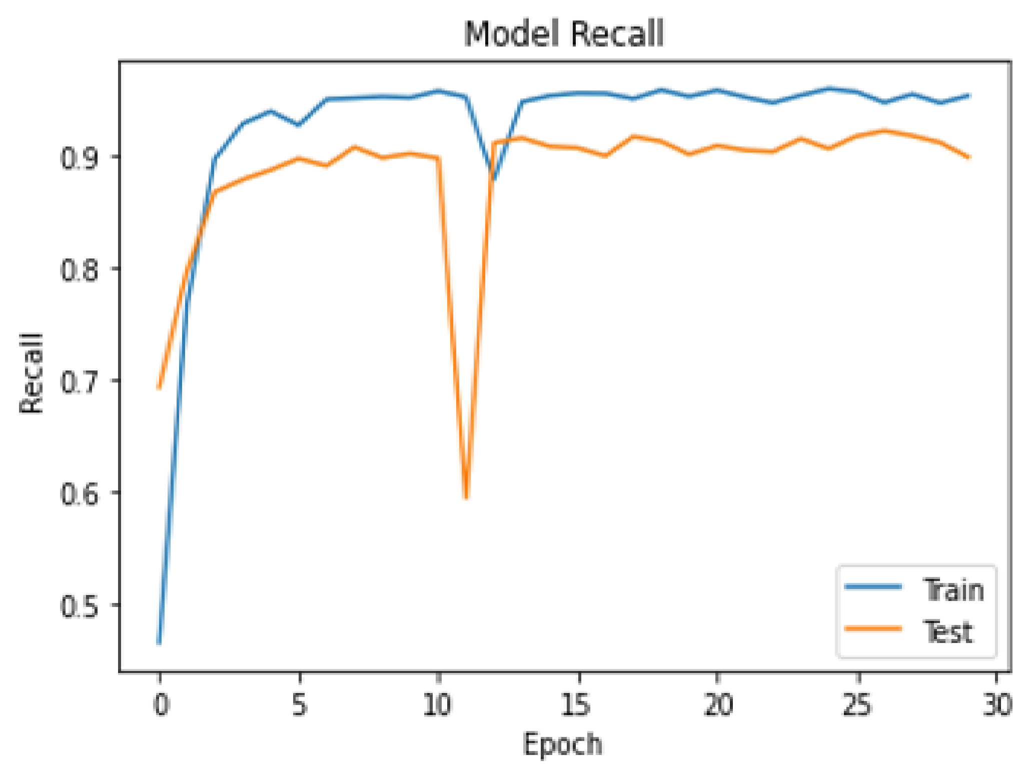
(a) Loss over epochs



(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.

Metrics Comparison between TPU and CPU

Model	Accuracy	Precision	Recall	F1 Score
TPU-DDL	89.92	90.22	89.85	90.04
CPU-DL	92.40	92.49	92.33	92.33

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!