

# ASSIGNMENT 4

OM SHRI PRASATH

EE17B113

## Introduction:

The assignment consists of creating a neural network using Tensorflow and training it on the New York City Taxi Fare Prediction dataset. We will train the dataset on CPU, GPU and TPU for different batch size given for the Batch Gradient Descent Process and find the relation between the time taken and the batch size. we will then train the model using different types of neural networks and find the relation with respect to error and training time.

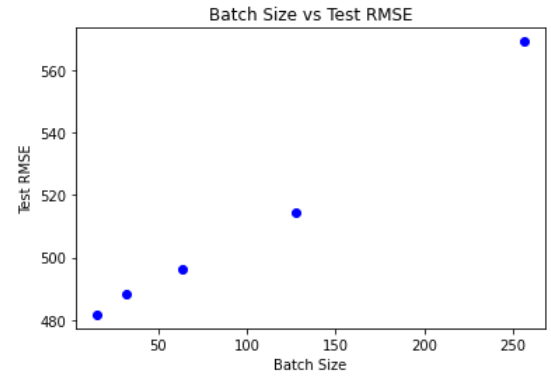
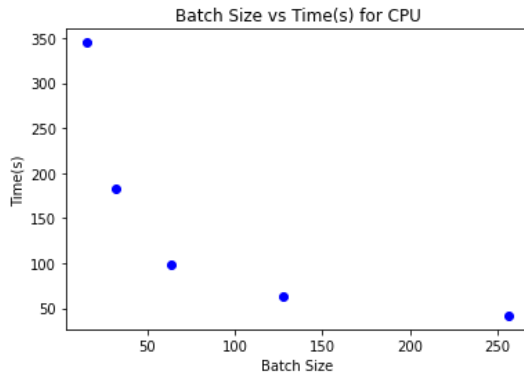
## Training using different Processors:

We will plot the time required for training vs the batch size used for training.

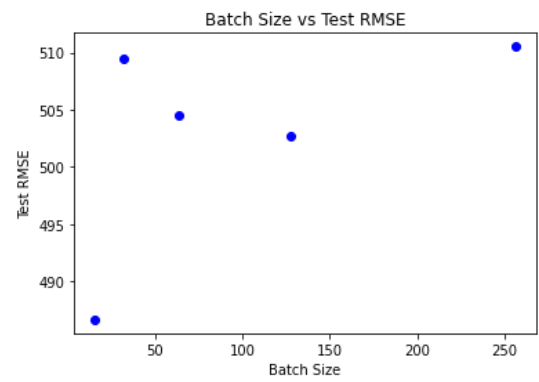
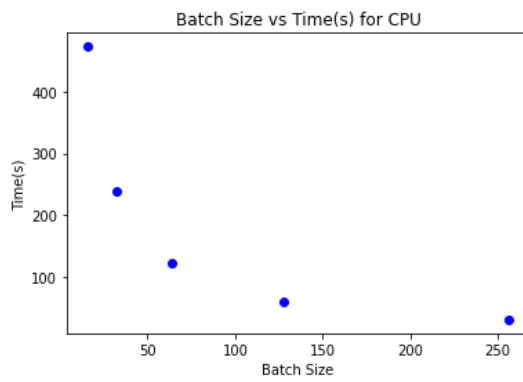
- **Training data size** = 86348
- **Test data size** = 21587
- **Input Shape** : (<batch\_size>,60)
- **Output Shape** : (1,)
- **Hidden Layers Nodes:**
  - Number of nodes in hidden layer : **[128,64,32,16,8]**
    - **Layer 0 (Input Layer)** : Input Size = 60, Output Size = 128
    - **Layer 1** : Input Size = 128, Output Size = 64
    - **Layer 2** : Input Size = 64, Output Size = 32
    - **Layer 3** : Input Size = 32, Output Size = 16
    - **Layer 4** : Input Size = 16, Output Size = 8
    - **Layer 5 (Output Layer)** : Input Size = 8, Output Size = 1
- **Number of epochs** = 50
- **Minibatch Size** = 16 , 32 , 64 , 128 , 256 , 512
- **Learning Rate** = 0.005

The results are as follows :

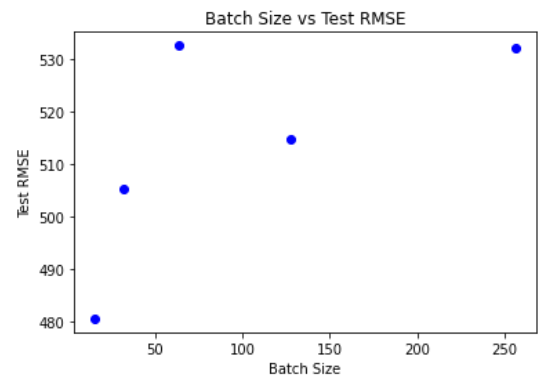
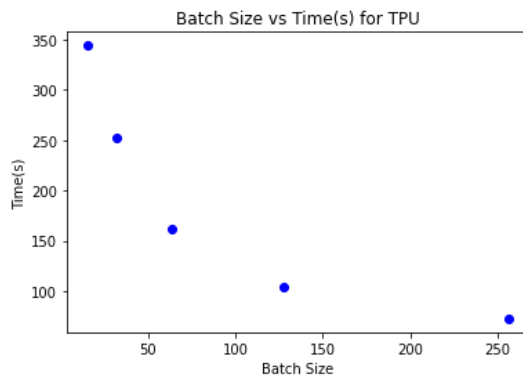
- CPU -



- GPU -



- TPU -



## Inferences on Training in CPU, GPU and TPU

- The training time with CPU, GPU and TPU has been plotted above.
- For smaller batch sizes, CPU has better performance, but as batch size increases, GPU becomes the better performer due to the increase in speed by vectorization overcoming slowness due to looping epoch execution.
- TPU performance is average overall since it has been designed for handling huge volumes of data.

## Neural Network Variation

- **Training is done using CPU using batch size of 64 for best time-accuracy compensation.**
- **Training data size = 86348**
- **Test data size = 21587**
- The number of nodes in the neural network to be tested are :
  - [64,32,16,8]
    - **Layer 0 (Input Layer)** : Input Size = 60, Output Size = 64
    - **Layer 1** : Input Size = 64, Output Size = 32
    - **Layer 2** : Input Size = 32, Output Size = 16
    - **Layer 3** : Input Size = 16, Output Size = 8
    - **Layer 4 (Output Layer)** : Input Size = 8, Output Size = 1
  - [64,128,256,128,64,32,16,8]
    - **Layer 0 (Input Layer)** : Input Size = 60, Output Size = 64
    - **Layer 1** : Input Size = 64, Output Size = 128
    - **Layer 2** : Input Size = 128, Output Size = 256
    - **Layer 3** : Input Size = 256, Output Size = 128
    - **Layer 4** : Input Size = 128, Output Size = 64
    - **Layer 5** : Input Size = 64, Output Size = 32
    - **Layer 6** : Input Size = 32, Output Size = 16
    - **Layer 7** : Input Size = 16, Output Size = 8
    - **Layer 8 (Output Layer)** : Input Size = 8, Output Size = 1

- [64,128,64,32,16,8]
  - **Layer 0 (Input Layer)** : Input Size = 60, Output Size = 64
  - **Layer 1** : Input Size = 64, Output Size = 128
  - **Layer 2** : Input Size = 64, Output Size = 32
  - **Layer 3** : Input Size = 32, Output Size = 16
  - **Layer 4** : Input Size = 16, Output Size = 8
  - **Layer 5 (Output Layer)** : Input Size = 8, Output Size = 1
- [32,8,1]
  - **Layer 0 (Input Layer)** : Input Size = 60, Output Size = 32
  - **Layer 1** : Input Size = 32, Output Size = 8
  - **Layer 2 (Output Layer)** : Input Size = 8, Output Size = 1

Hidden Layers	Time Per Epoch(s)	Test RMSE
[64,32,16,8]	0.8176876020431518	483.59985
[64,128,256,128,64,32,16,8]	1.8521811413764953	628.48413
[64,128,64,32,16,8]	1.1637224793434142	492.48026
[32,8,1]	0.6738513231277465	574.79034

From the table, we can infer the following results :

- Time per epoch increases as the number of nodes in the neural network increases. This is simply because the number of computations per epoch increases with an increase in the neural network size.
- The relation between accuracy with neural network size is somewhat hard to explain, but as a general rule, too small a network or too large a network will have poor accuracy due to bias and variance problem respectively. So finding the best possible size for the neural network involves slowly changing the network size and checking the accuracy of the network using a small subset of the data, and tuning the model hyperparameters along with it.

## Result :

Thus we have plotted the time taken for execution with different types of processors (CPU, GPU and TPU) and have tried different neural networks and found the Test RMSE and time taken per epoch with the explanation.