# Sequence Model Computations speed-up using Multithreading for Sentiment Review Classification on AWS Cloud Resources

ABHISHEK BABUJI

# LITERATURE SURVEY

1. Large Scale Distributed Deep Networks

Helps understand how parallelization setting is applied on the data level and on the model level

2. Conflict-free Asynchronous Machine Learning

Helps understand how results from mini-batches are gathered and aggregated in a conflict free manner

# THE EXPERIMENT AND MOTIVATIONS

# THE EXPERIMENT

- Kaggle Amazon Fine Food Reviews: https://www.kaggle.com/snap/amazon-fine-food-reviews
- Data divided into:
  1. Good reviews: 125323
  2. Bad reviews: 44896
  3. Average reviews: 15430

# THE MOTIVATION

- Compare Naive Bayes to Sequence Models in Natural Language Processing Task
- Understand the scope of improvement that Sequence Models can provide
- Understand the setting required to achieve better results in shorter time using Multithreading

# NAIVE BAYES

## THE BEST THAT NAIVE BAYES CAN DO: TOTAL – 36 EXPERIMENTS

- 36 different combinations of the same dataset are created

  1. Stemming/Lemmatization/ Default - 3 options

  2. Keep/Remove Stop Words - 2 options

  3. TF/TF-IDF as feature weighting scheme

  4. N-gram ranges: Uni/Bi/Uni and Bi - 3 options

| Stem, Lemmatize, Default | Stop words | Feature weighting Scheme | n-gram range | Training accuracy | Test accuracy |
|---|---|---|---|---|---|
| N/A | N/A | TF | bi | 86.4% | 87.2% |
| Lemmatize | N/A | TF | bi | 86.2% | 87.3% |
| Stem | N/A | TF | bi | 86.2% | 87.0% |
| Stem | N/A | TF | uni and bi | 85.7% | 86.5% |

# SEQUENCE MODELS - LSTM

## THE BEST THAT SEQUENCE MODELS CAN DO: TOTAL – 16 EXPERIMENTS

- Hardware: 16 vCPUs, 32 GiB RAM - AWS
- Number of Epochs: 100
- Architecture
  1. Dropout
  2. 1D Temporal Activation
  3. A max-pooling layer
  4. LSTM units - 100
  5. Dense softmax activation

| Batch size | Number of units | Training accuracy | Test accuracy |
|---|---|---|---|
| 32 | 100 | 86.2% | 85.6% |
| 1024 | 100 | 92.1% | 86.3% |

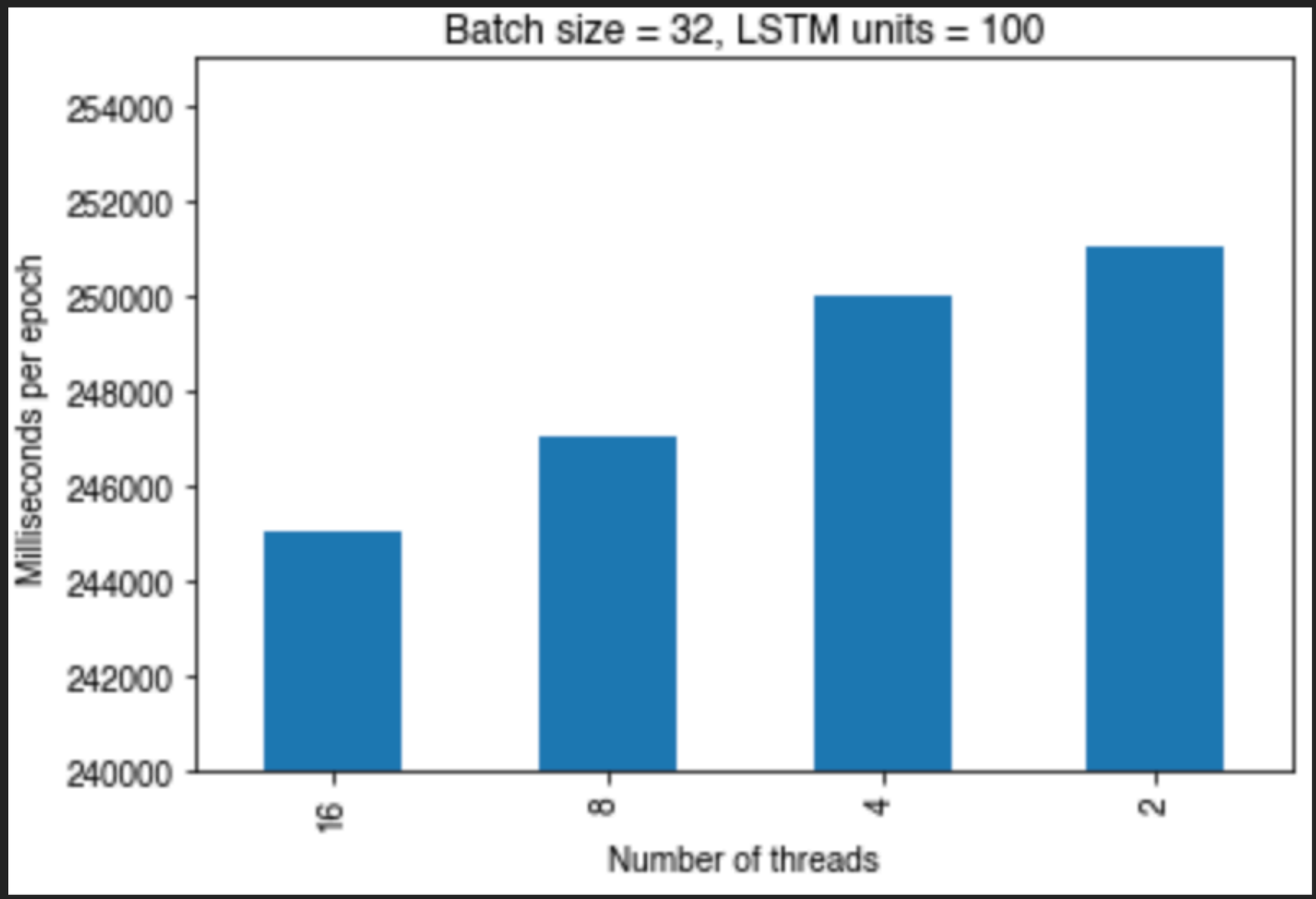
# MULTITHREADING SETTING RESULTS

| Batch size | Number of threads | Number of units | Average time per epoch |
|---|---|---|---|
| 32 | 16 | 100 | 4m 5s |
| 32 | 16 | 1000 | 18m 7s |
| 32 | 8 | 100 | 4m 7s |
| 32 | 8 | 1000 | 18m 13s |
| 32 | 4 | 100 | 4m 10s |
| 32 | 4 | 1000 | 18m 13s |
| 32 | 2 | 100 | 4m 11s |
| 32 | 2 | 1000 | 18m 16s |
| 1024 | 16 | 100 | 55s |
| 1024 | 16 | 1000 | 3m 57s |
| 1024 | 8 | 100 | 57s |
| 1024 | 8 | 1000 | 3m 58s |
| 1024 | 4 | 100 | 59s |
| 1024 | 4 | 1000 | 3m 59s |
| 1024 | 2 | 100 | 59s |
| 1024 | 2 | 1000 | 4m |

# RESULTS AT A CLOSER GLANCE

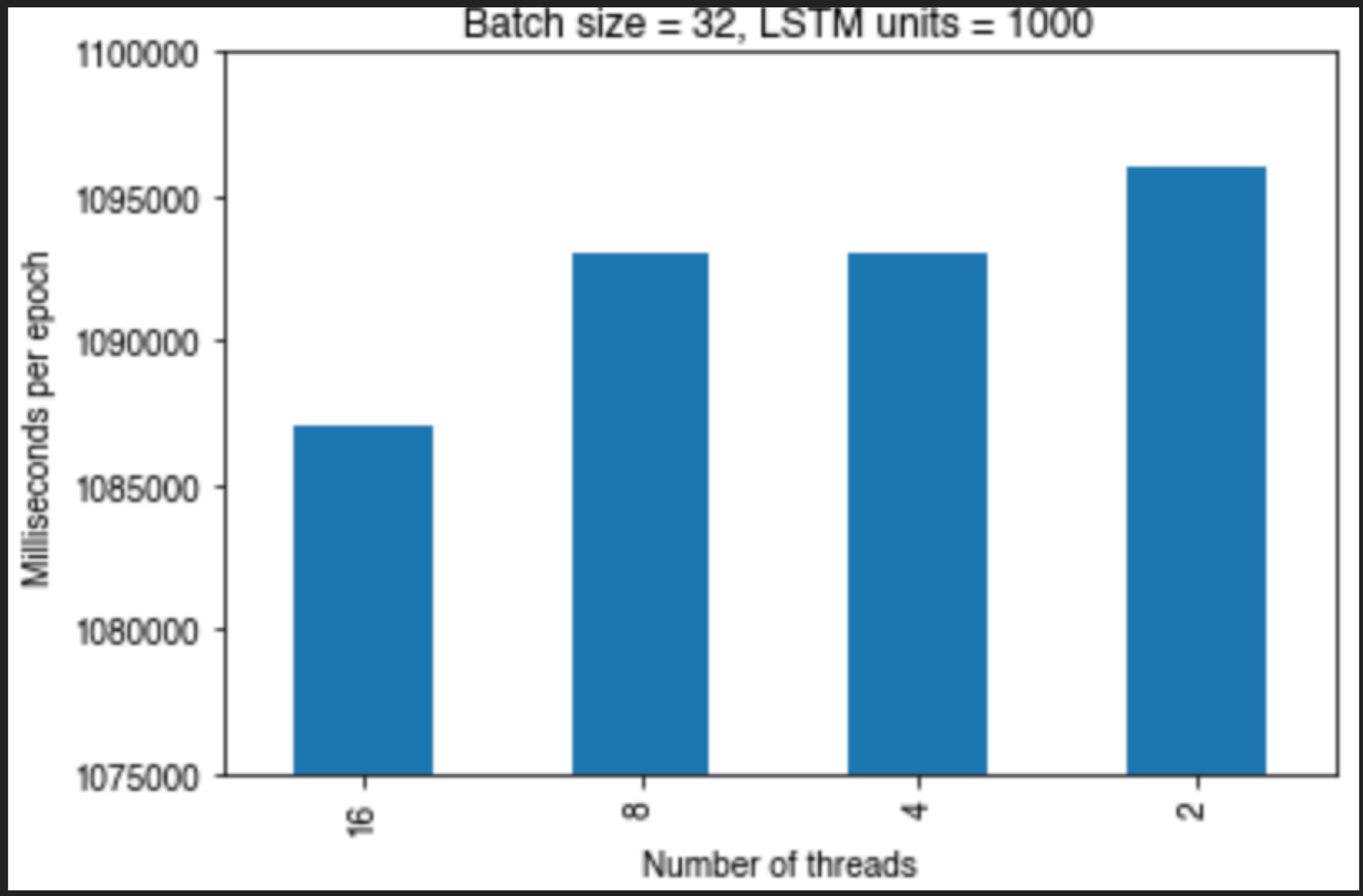
## BATCH SIZE = 32, NUMBER OF UNITS = 100

| | batch_size | num_threads | num_units | time_per_epoch_in_milliseconds |
|---|---|---|---|---|
| 0 | 32 | 16 | 100 | 245000.0 |
| 2 | 32 | 8 | 100 | 247000.0 |
| 4 | 32 | 4 | 100 | 250000.0 |
| 6 | 32 | 2 | 100 | 251000.0 |



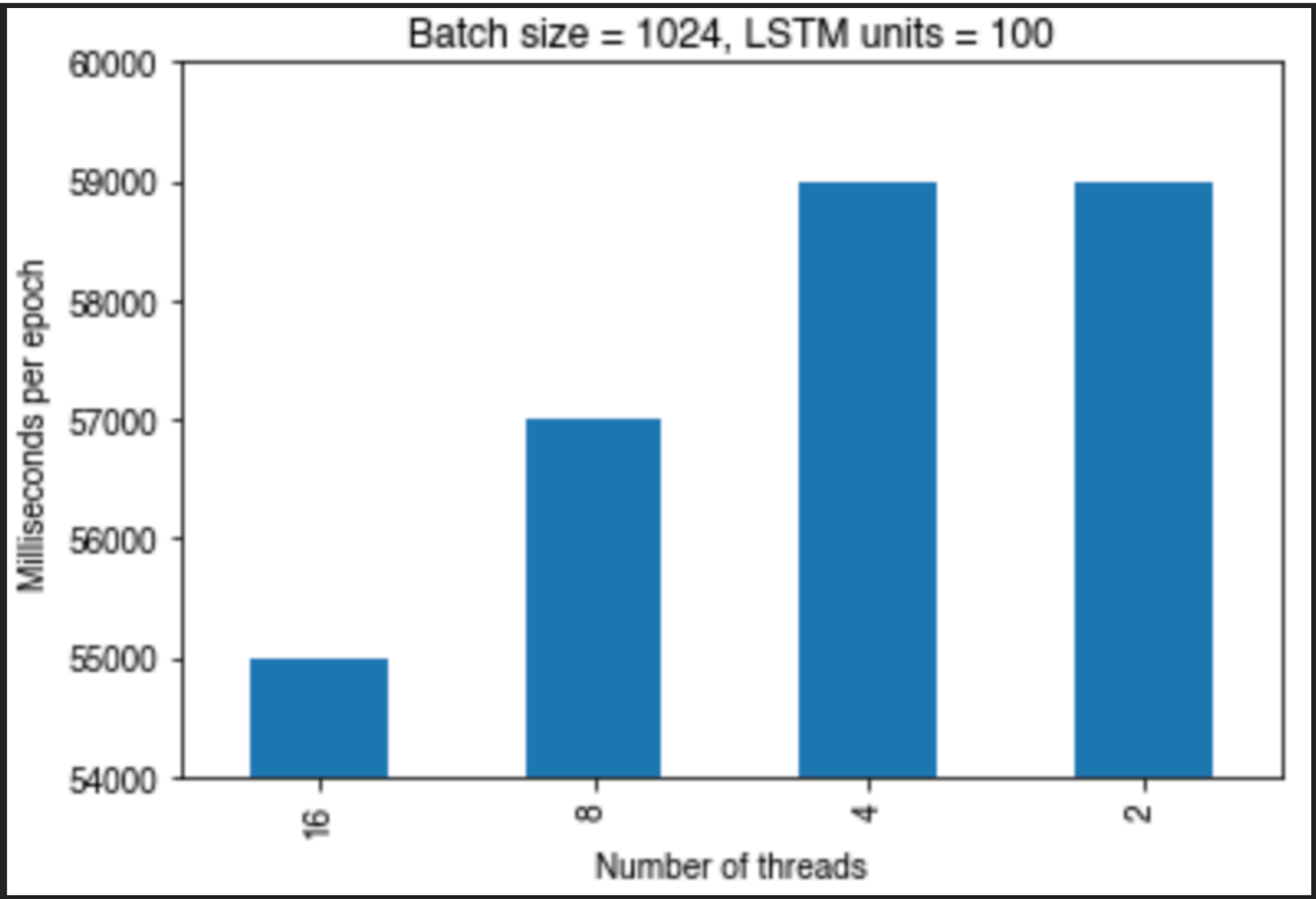
## BATCH SIZE = 32, NUMBER OF UNITS = 1000

| | batch_size | num_threads | num_units | time_per_epoch_in_milliseconds |
|---|---|---|---|---|
| 1 | 32 | 16 | 1000 | 1087000.0 |
| 3 | 32 | 8 | 1000 | 1093000.0 |
| 5 | 32 | 4 | 1000 | 1093000.0 |
| 7 | 32 | 2 | 1000 | 1096000.0 |



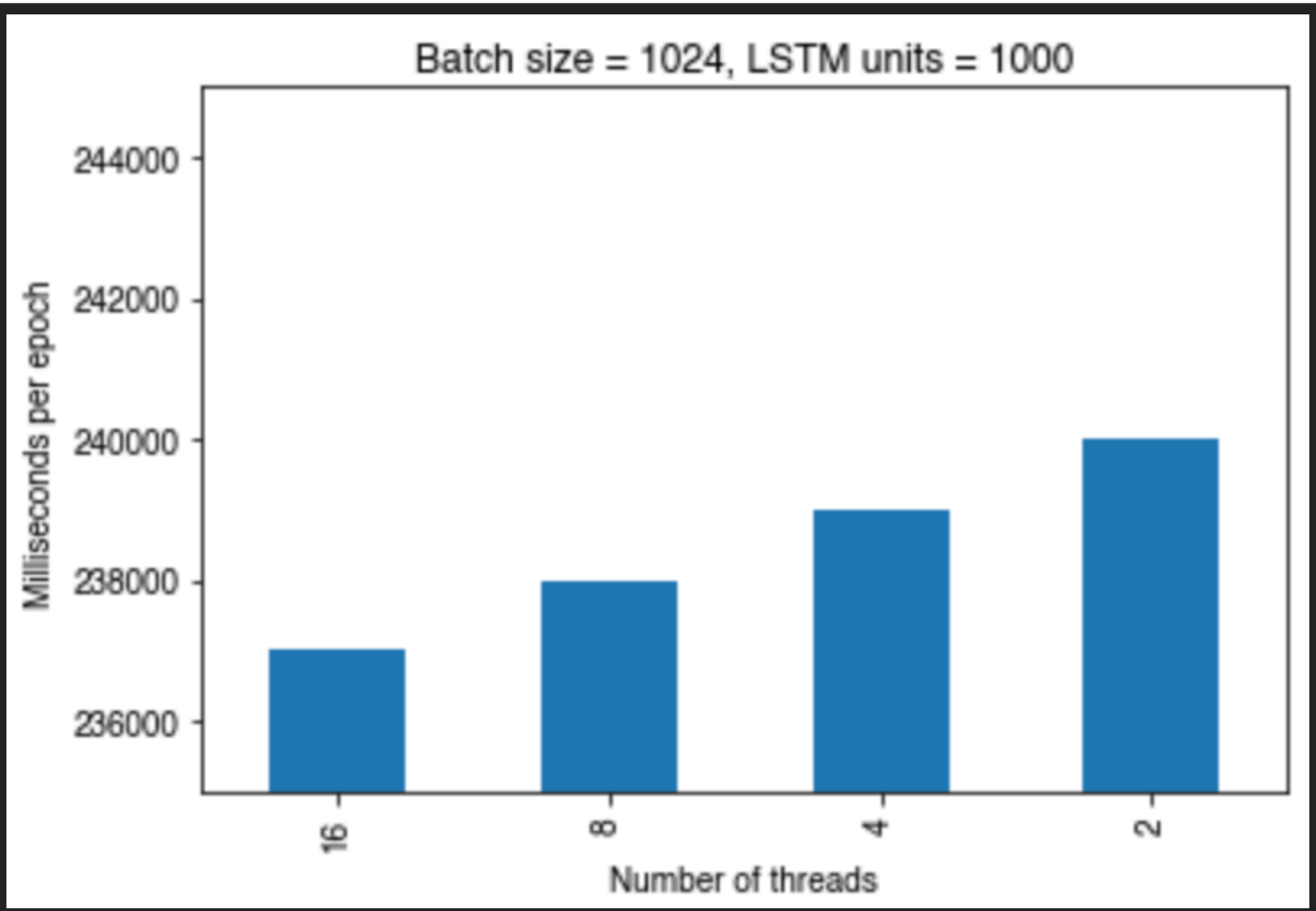
# BATCH SIZE = 1024, NUMBER OF UNITS = 100

| | batch_size | num_threads | num_units | time_per_epoch_in_milliseconds |
|---|---|---|---|---|
| 8 | 1024 | 16 | 100 | 55000.0 |
| 10 | 1024 | 8 | 100 | 57000.0 |
| 12 | 1024 | 4 | 100 | 59000.0 |
| 14 | 1024 | 2 | 100 | 59000.0 |



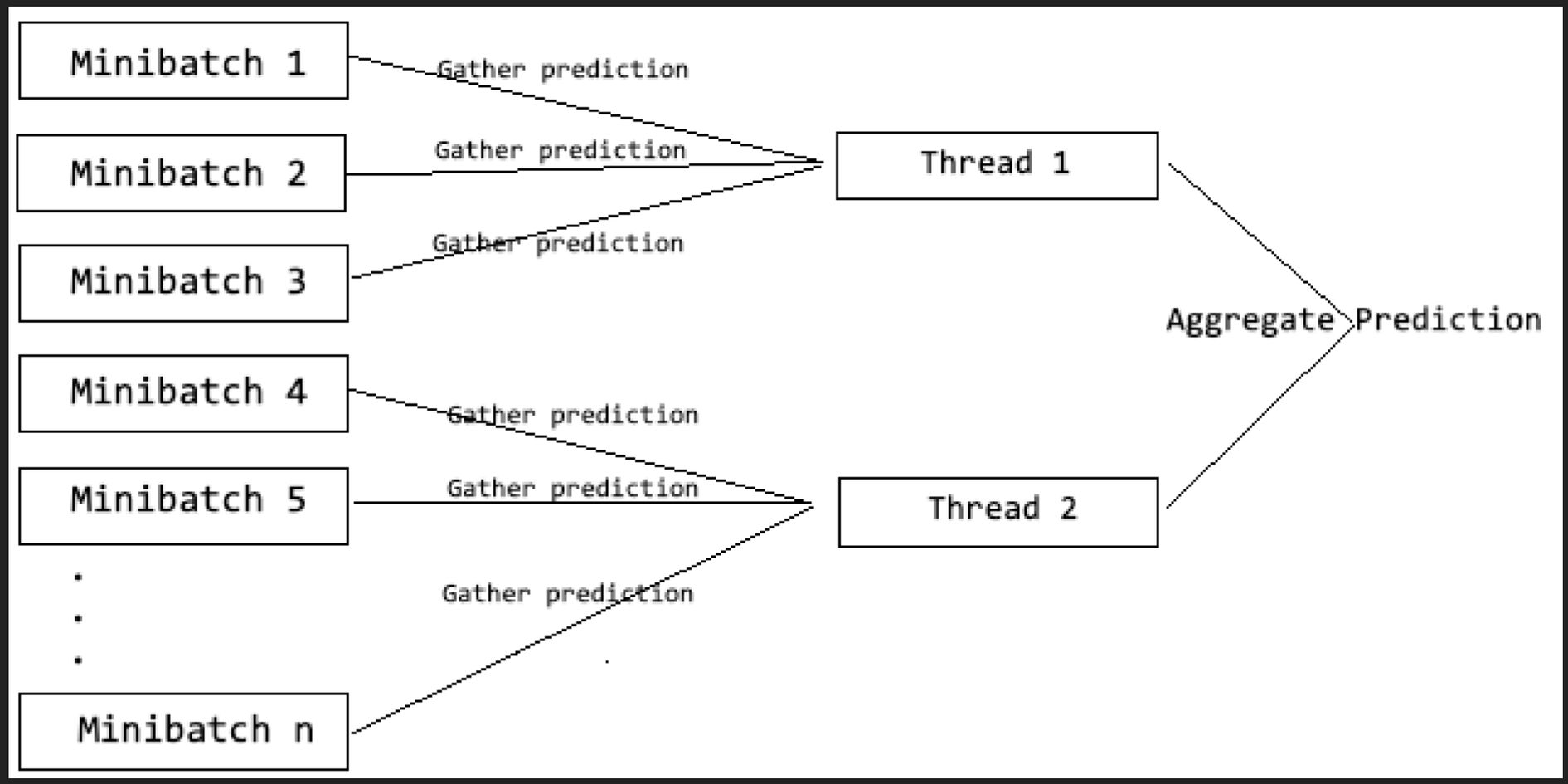
## BATCH SIZE = 1024, NUMBER OF UNITS = 1000

| | batch_size | num_threads | num_units | time_per_epoch_in_milliseconds |
|---|---|---|---|---|
| 9 | 1024 | 16 | 1000 | 237000.0 |
| 11 | 1024 | 8 | 1000 | 238000.0 |
| 13 | 1024 | 4 | 1000 | 239000.0 |
| 15 | 1024 | 2 | 1000 | 240000.0 |



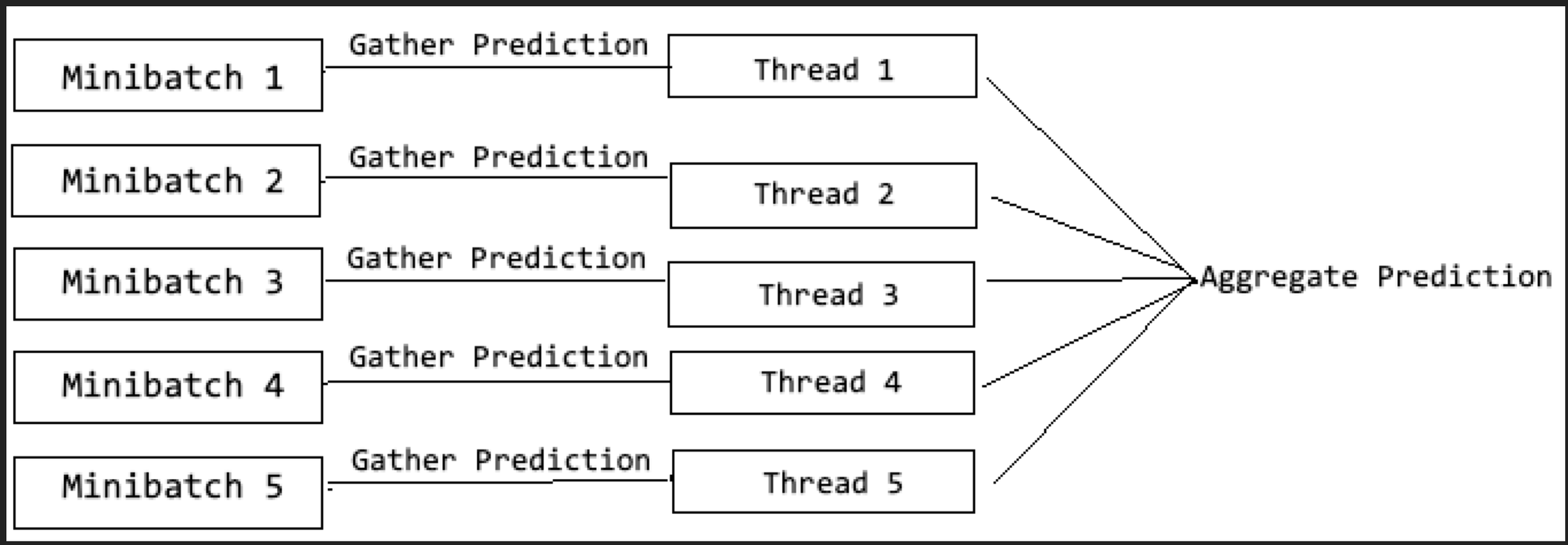
# OBSERVATIONS

# EFFECT OF FEWER THREADS




- Sequential gathering of predictions by each thread.
- Lesser the number of threads, more the time taken to gather predictions and hence increased time taken to train

# EFFECT OF MORE THREADS




- More parallelized gathering of predictions by each thread.
- More the number of threads, lesser the time taken to gather predictions and hence decreased time taken to train

# CONCLUSIONS

## CONCLUSIONS

- SEQUENCE MODELS PROVIDE BETTER ACCURACY THAN NAIVE BAYES
- CREATING MORE NUMBER OF BATCHES INCREASE THE TIME TAKEN FOR GRADIENT DESCENT CONVERGENCE
- MAXIMUM THREAD USAGE APPEARS TO BE CAPPED AT ONE HALF OF THE NUMBER OF vCPUs
- WHEN LESSER NUMBER OF THREADS ARE BURDENED WITH MORE MINI-BATCHES, GATHERING AND AGGREGATION OF THE RESULTS OF MINI-BATCH TAKES A LOT OF TIME
- WHEN THERE ARE LESSER NUMBER OF BATCHES (SIZE OF EACH BATCH BEING LARGE) GRADIENT DESCENT TAKES SIGNIFICANTLY LESSER TIME TO CONVERGE