

Google Discussions



Exam Professional Machine Learning Engineer All Questions

View all questions & answers for the Professional Machine Learning Engineer exam

Go to Exam

EXAM PROFESSIONAL MACHINE LEARNING ENGINEER TOPIC 1 QUESTION 82 DISCUSSIO..

Actual exam question from Google's Professional Machine Learning Engineer

Question #: 82

Topic #: 1

[\[All Professional Machine Learning Engineer Questions\]](#)

You are profiling the performance of your TensorFlow model training time and notice a performance issue caused by inefficiencies in the input data pipeline for a single 5 terabyte CSV file dataset on Cloud Storage. You need to optimize the input pipeline performance. Which action should you try first to increase the efficiency of your pipeline?

- A. Preprocess the input CSV file into a TFRecord file.
- B. Randomly select a 10 gigabyte subset of the data to train your model.
- C. Split into multiple CSV files and use a parallel interleave transformation.
- D. Set the `reshuffle_each_iteration` parameter to true in the `tf.data.Dataset.shuffle` method.

Show Suggested Answer

by [LearnSodas](#) at Dec. 11, 2022, 6:03 p.m.

Comments

Type your comment...

Submit

[Prakzz](#) 3 months, 3 weeks ago

Selected Answer: A

Preprocessing the input CSV file into a TFRecord file optimizes the input data pipeline by enabling more efficient reading and processing. TFRecord is a binary format that is faster to read and more efficient for TensorFlow to process compared to CSV, which is a text-based format. This change can significantly reduce the time spent on data input operations during model training.

   upvoted 3 times

  **PhilipKoku** 4 months, 2 weeks ago

Selected Answer: A

A) Convert CSV file into TFRecord is more efficient and processing CSV in parallel (C)

   upvoted 1 times

  **pinimichele01** 6 months ago

Selected Answer: C

Converting a large 5 terabyte CSV file to a TFRecord can be a time-consuming process, and you would still be dealing with a single large file.

   upvoted 2 times

  **tavva_prudhvi** 11 months, 2 weeks ago

Selected Answer: C

While preprocessing the input CSV file into a TFRecord file (Option A) can improve the performance of your input pipeline, it is not the first action to try in this situation. Converting a large 5 terabyte CSV file to a TFRecord can be a time-consuming process, and you would still be dealing with a single large file.

   upvoted 1 times

  **andresvelasco** 1 year, 1 month ago

Selected Answer: C

I think C based on the consideration: "Which action should you try first", meaning it should be less impactful to continue using CSV.

   upvoted 1 times

  **TNT87** 1 year, 4 months ago

Selected Answer: C

https://www.tensorflow.org/guide/data_performance#best_practice_summary

   upvoted 2 times

  **M25** 1 year, 5 months ago

Selected Answer: C

Went with C

   upvoted 1 times

  **e707** 1 year, 5 months ago

Selected Answer: C

Option A, preprocess the input CSV file into a TFRecord file, is not as good because it requires additional processing time. Hence, I think C is the best choice.

   upvoted 1 times

  **frangm23** 1 year, 6 months ago

Selected Answer: A

I think it could be A.

https://cloud.google.com/architecture/best-practices-for-ml-performance-cost#preprocess_the_data_once_and_save_it_as_a_tfrecord_file

   upvoted 1 times

  **[Removed]** 1 year, 6 months ago

Selected Answer: A

Clearly both A and C works here, but I can't find any documentation which suggests C is any better than A.



   upvoted 1 times

  **Yajnas_arpohc** 1 year, 7 months ago

"Which action should you try first" seems to be key -- C seems more intuitive as first step!

A is valid as well (interleave works w TFRecords) & definitely more efficient IMO, but maybe 2nd step!

   upvoted 2 times

  **shankalman717** 1 year, 8 months ago

Selected Answer: A

Option B (randomly selecting a 10 gigabyte subset of the data) could lead to a loss of useful data and may not be representative of the entire dataset. Option C (splitting into multiple CSV files and using a parallel interleave transformation)

may also improve the performance, but may be more complex to implement and maintain, and may not be as efficient as converting to TFRecord. Option D (setting the `reshuffle_each_iteration` parameter to true in the `tf.data.Dataset.shuffle` method) is not directly related to the input data format and may not provide as significant a performance improvement as converting to TFRecord.

   upvoted 3 times

  **tavva_prudhvi** 1 year, 7 months ago

Please read this site https://www.tensorflow.org/tutorials/load_data/csv, its simple to implement in the same input pipeline, and we cannot judge the answer by implementation difficulties!

   upvoted 1 times



  **SMASL** 1 year, 8 months ago

Could anyone be kind to explain why C is preferred over A? My initial guess was on A, but everyone here seems to unanimously prefer C. Is it because it is not about optimizing I/O performance, but rather the input `_pipeline_`, which is about processing arrived data within that TF input pipeline (non-I/O)? I just try to understand here. Thanks for reply in advance!

   upvoted 4 times

  **tavva_prudhvi** 1 year, 7 months ago

Option C, splitting into multiple CSV files and using a parallel interleave transformation, could improve the pipeline efficiency by allowing multiple workers to read the data in parallel.

   upvoted 1 times

  **[Removed]** 1 year, 6 months ago

yes but how is it more efficient than converting to a TFRecord file?

   upvoted 1 times

  **tavva_prudhvi** 1 year, 3 months ago

A TFRecord file is a binary file format that is used to store TensorFlow data. It is more efficient than a CSV file because it can be read more quickly and it takes up less space. However, it is still a large file, and it would take a long time to read it into memory. Splitting the file into multiple smaller files would reduce the amount of time it takes to read the files into memory, and it would also make it easier to parallelize the reading process.

   upvoted 1 times

  **enghabeth** 1 year, 8 months ago

Selected Answer: C

split data it's best way in my opinion



   upvoted 1 times

  **hiromi** 1 year, 10 months ago

Selected Answer: C


C
Keywords -> You need to optimize the input pipeline performance
https://www.tensorflow.org/guide/data_performance

   upvoted 2 times

  **hiromi** 1 year, 10 months ago

- https://www.tensorflow.org/tutorials/load_data/csv

   upvoted 1 times

  **ares81** 1 year, 10 months ago

Selected Answer: C

It seems C, to me.

   upvoted 1 times

  **LearnSodas** 1 year, 10 months ago

Selected Answer: C

Splitting the file we can use parallel interleave to parallel load the datasets
https://www.tensorflow.org/guide/data_performance

   upvoted 2 times

Start Learning for free



Social Media

[Facebook](#) , [Twitter](#)

[YouTube](#) , [Reddit](#)

[Pinterest](#)



We are the biggest and most updated IT certification exam material website.

Using our own resources, we strive to strengthen the IT professionals community for free.



© 2024 ExamTopics

ExamTopics doesn't offer Real Microsoft Exam Questions. ExamTopics doesn't offer Real Amazon Exam Questions. ExamTopics Materials do not contain actual questions and answers from Cisco's Certification Exams.

CFA Institute does not endorse, promote or warrant the accuracy or quality of ExamTopics. CFA® and Chartered Financial Analyst® are registered trademarks owned by CFA Institute.