

EXAMTOPICS

- Expert Verified, Online, Free.

≡ MENU



🔍 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 135 DISCUSSI...

Actual exam question from Google's Professional Machine Learning Engineer

Question #: 135

Topic #: 1

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

You work for the AI team of an automobile company, and you are developing a visual defect detection model using TensorFlow and Keras. To improve your model performance, you want to incorporate some image augmentation functions such as translation, cropping, and contrast tweaking. You randomly apply these functions to each training batch. You want to optimize your data processing pipeline for run time and compute resources utilization. What should you do?

- A. Embed the augmentation functions dynamically in the tf.Data pipeline.
- B. Embed the augmentation functions dynamically as part of Keras generators.
- C. Use Dataflow to create all possible augmentations, and store them as TFRecords.
- D. Use Dataflow to create the augmentations dynamically per training run, and stage them as TFRecords.

Show Suggested Answer

by  mil_spyro at Dec. 13, 2022, 6:31 p.m.

Comments

Type your comment...

Submit

  **guilhermebutzke** 9 months, 1 week ago

Selected Answer: A

Option A: By embedding augmentation in the tf.data pipeline, data augmentation is applied on-the-fly during training, reducing the need to store pre-augmented data. Option B could be a choice, but since Keras generators are built on top of tf.data, they are less flexible and have a lower level of optimization compared to tf.data.

   upvoted 3 times

  **tavva_prudhvi** 11 months, 4 weeks ago

Selected Answer: A

A is best because,

1. It allows you to apply the augmentations on-the-fly during training, which eliminates the need for pre-processing and storing a large number of augmented images. This saves both storage space and compute resources.
2. The tf.Data pipeline is highly optimized for efficient data loading and processing, ensuring that your model training process is not bottlenecked by data preprocessing.
3. By applying augmentations randomly to each training batch, you increase the diversity of your training data, which can help your model generalize better to unseen data.

Keras generators can be used for data augmentation, but tf.Data pipelines are generally more efficient and flexible for creating complex data processing pipelines.

   upvoted 2 times

  **pico** 11 months, 4 weeks ago

Selected Answer: B

B (but also A)

B is a common and efficient approach for applying data augmentation during training. This allows you to apply data augmentation on-the-fly without the need to pre-generate or store augmented images separately, which saves storage space and reduces the preprocessing time. Keras provides various tools and functions for data augmentation, and you can easily incorporate them into your training data pipeline.

A can also be a good choice, especially if you are using TensorFlow's tf.data API for data loading and preprocessing. It can provide similar benefits by applying augmentations on-the-fly, but it may require more custom code to implement compared to Keras data generators.

   upvoted 1 times

  **guilhermebutzke** 9 months, 1 week ago

Yes, but I think the questions says: "You want to optimize your data processing pipeline for run time and compute resources utilization". keras is not optimized as tf.Data

   upvoted 1 times

  **envest** 1 year, 2 months ago

by abylead: B) Keras generators embedded augmentation functions offers at least Translation, Crop, and Contrast preprocessing. You can either permanently integrate the functions or you randomly use a dataset with non CPU blocking async training batches & optimized GPU processing overlapping. By applying Keras embedded augmentation functions, the tf.data pipeline can still be performance optimized.

With tf.image pipelines you lack pipeline performance optimization & the deprecated translation function. In addition, the complex application hinders random operation flexibility.

   upvoted 1 times

  **PST21** 1 year, 3 months ago

B - TensorFlow's Keras API provides built-in support for data augmentation using various image preprocessing layers, such as RandomTranslation, RandomCrop, and RandomContrast, among others. You can create custom image augmentation functions and include them as part of your Keras generators, tailoring them to your specific use case and needs.

In summary, Option B, embedding the augmentation functions dynamically as part of Keras generators, offers efficient on-the-fly data augmentation, reduced storage overhead, optimized resource utilization, and greater flexibility, making it the best choice for this scenario.

   upvoted 1 times

  **tavva_prudhvi** 1 year, 3 months ago

lthough Keras generators can be used for data augmentation, using the tf.data pipeline provides better performance and efficiency. The tf.data API is more flexible and better integrated with TensorFlow, allowing for more optimizations.especially if you have a large number of images to process.

   upvoted 2 times

  **M25** 1 year, 5 months ago

Selected Answer: A

Went with A

   upvoted 1 times

matamata415 1 year, 7 months ago

Selected Answer: A

[https://www.tensorflow.org/tutorials/load_data/images?](https://www.tensorflow.org/tutorials/load_data/images?hl=ja#tfdata_%E3%82%92%E4%BD%BF%E7%94%A8%E3%81%97%E3%81%A6%E3%82%88%E3%82%8A%E7%B2%BE%E5%AF%86%E3%81%AB%E5%88%B6%E5%BE%A1%E3%81%99%E3%82%8B)

[hl=ja#tfdata_%E3%82%92%E4%BD%BF%E7%94%A8%E3%81%97%E3%81%A6%E3%82%88%E3%82%8A%E7%B2%BE%E5%AF%86%E3%81%AB%E5%88%B6%E5%BE%A1%E3%81%99%E3%82%8B](https://www.tensorflow.org/tutorials/load_data/images?hl=ja#tfdata_%E3%82%92%E4%BD%BF%E7%94%A8%E3%81%97%E3%81%A6%E3%82%88%E3%82%8A%E7%B2%BE%E5%AF%86%E3%81%AB%E5%88%B6%E5%BE%A1%E3%81%99%E3%82%8B)

upvoted 2 times

matamata415 1 year, 7 months ago

https://www.tensorflow.org/tutorials/load_data/images#using_tfdata_for_finer_control

upvoted 2 times

Yajnas_arpohc 1 year, 7 months ago

Selected Answer: A

<https://towardsdatascience.com/time-to-choose-tensorflow-data-over-imagedatagenerator-215e594f2435>

upvoted 1 times

TNT87 1 year, 8 months ago

Selected Answer: A

A. Embed the augmentation functions dynamically in the tf.Data pipeline is the best approach to optimize the data processing pipeline for runtime and compute resource utilization.

Using the tf.data pipeline, you can apply data augmentation functions dynamically to each batch during training. This approach avoids the overhead of creating preprocessed TFRecords or Keras generators, which can consume additional disk space, memory, and CPU. Additionally, using the tf.data pipeline, you can parallelize data preprocessing, input pipeline operations, and model training

upvoted 2 times

shankalman717 1 year, 8 months ago

Selected Answer: A

Embedding the augmentation functions dynamically in the tf.Data pipeline allows the data pipeline to apply the augmentations on the fly as the data is being loaded into the model during training. This means that the model can utilize the compute resources effectively by loading and processing the data as needed, rather than pre-generating all possible augmentations ahead of time (as in options C and D), which could be computationally expensive and time-consuming.

Option B is also a viable choice, but it may not be as efficient as option A since the data augmentation functions would be applied during training using Keras generators, which could cause some overhead.

upvoted 2 times

pshemol 1 year, 9 months ago

Selected Answer: B

will go for B too

<https://www.analyticsvidhya.com/blog/2020/08/image-augmentation-on-the-fly-using-keras-imagedatagenerator/>

upvoted 1 times

John_Pongthorn 1 year, 9 months ago

Either of A or B : I am not convinced of what the right answer is. but it is on

https://www.tensorflow.org/tutorials/images/data_augmentation#apply_augmentation_to_a_dataset certainly

upvoted 1 times

hiromi 1 year, 10 months ago

Selected Answer: A

A (not sure)

upvoted 1 times

YangG 1 year, 10 months ago

Selected Answer: B

will go for B

<https://stanford.edu/~shervine/blog/keras-how-to-generate-data-on-the-fly>

upvoted 2 times

mil_spyro 1 year, 10 months ago

Selected Answer: A

incorporating the augmentation functions into the pipeline, you can apply them dynamically to each training batch, without the need to generate all possible augmentations in advance or stage them as TFRecords.

upvoted 4 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.