EXAM PROFESSIONAL MACHINE LEARNING ENGINEER TOPIC 1 QUESTION 96 DISCUSSIO..

Actual exam question from Google's Professional Machine Learning Engineer
Question #: 96
Topic #: 1
[All Professional Machine Learning Engineer Questions]

You are training an object detection machine learning model on a dataset that consists of three million X-ray images, each roughly 2 GB in size. You are using Vertex AI Training to run a custom training application on a Compute Engine instance with 32-cores, 128 GB of RAM, and 1 NVIDIA P100 GPU. You notice that model training is taking a very long time. You want to decrease training time without sacrificing model performance. What should you do?

A. Increase the instance memory to 512 GB and increase the batch size.

B. Replace the NVIDIA P100 GPU with a v3-32 TPU in the training job.

C. Enable early stopping in your Vertex AI Training job.

D. Use the tf.distribute.Strategy API and run a distributed training job.

Show Suggested Answer

by hiromi at *Dec. 20, 2022, 12:08 a.m.*

Comments

smarques Highly Voted 1 year, 9 months ago

Selected Answer: C

I would say C.

The question asks about time, so the option "early stopping" looks fine because it will no impact the existent accuracy (it will maybe improve it).

The tf.distribute.Strategy reading the TF docs says that it's used when you want to split training between GPUs, but the question says that we have a single GPU.

Open to discuss. :)

upvoted 7 times

djo06 1 year, 3 months ago

tf.distribute.OneDeviceStrategy uses parallel training on one GPU

upvoted 2 times

Th3N1c3Guy Most Recent 1 month ago

Selected Answer: B

since compute engine is being used, seems like GPU upgrade makes sense

upvoted 1 times

baimus 1 month, 1 week ago

Selected Answer: D

The difficulty of this question is it's pure ambiguity. Two of the answer DO change the hardware, so this is obviously an option. The distribute strategy is clearly the right choice (D) assuming we are allowed more hardware to distribute it over. People are saying "we cannot change the hardware so it's B", but B is a change of hardware to TPU anyway, which would require a code change, at which point D would be implemented anyway.

upvoted 1 times

MultiCloudIronMan 1 month, 2 weeks ago

Selected Answer: D

I have seen two or even 3 of this question and there are strong debates on the answer, I want to suggest D, because Yes, distributed training can work with your setup of 32 cores, 128 GB of RAM, and 1 NVIDIA P100 GPU. However, the efficiency and performance will depend on the specific framework and strategy you use. The important thing about this answer is that it does not affect quality

upvoted 1 times

Jason_Cloud_at 1 month, 3 weeks ago

Selected Answer: B

in the question it says 3 Million xrays each with 2 GB , it will round upto 6M in size, TPU are exactly designed to accelerate ML tasks and it does massive parallelism, so i would go with B , i would directly omit A , C coz it is more about preventing and not directly aimed at reducing downtime, D is viable solution but comapring with B it is not.

upvoted 1 times

dija123 4 months ago

Selected Answer: B

Agree with B

upvoted 1 times

inc_dev_ml_001 6 months ago

Selected Answer: B

I would say B:

A. Increse memory doesn't mean necessary a speed up of the process, it's not a batch-size problem
B. It seems a image -> Tensorflow situation. So transforming image into tensors means that a TPU works better and maybe faster
C. It's not a overfitting problem
D. Same here, it's not a memory or input-size problem

upvoted 2 times

pinimichele01 6 months ago

https://www.tensorflow.org/guide/distributed_training#onedevicestrategy

upvoted 1 times

pinimichele01 6 months ago

https://www.tensorflow.org/guide/distributed_training#onedevicestrategy
-> D

upvoted 1 times

Werner123 7 months, 3 weeks ago

Selected Answer: D

In my eyes the only solution is distributed training. 3 000 000 x 2GB = 6 Petabytes worth of data. No single device will get you there.

upvoted 2 times

ludovikush 7 months, 3 weeks ago

Selected Answer: B

Agree with JamesDoes

upvoted 1 times

Mickey321 11 months, 1 week ago

Selected Answer: B

B as it have only one GPU hence in D distributed not efficient

upvoted 3 times

pico 11 months, 1 week ago

f the question didn't specify the framework used, and you want to choose an option that is more framework-agnostic, it's important to consider the available options.

Given the context and the need for a framework-agnostic approach, you might consider a combination of options A and D. Increasing instance memory and batch size can still be beneficial, and if you're using a deep learning framework that supports distributed training (like TensorFlow or PyTorch), implementing distributed training (Option D) can further accelerate the process.

upvoted 1 times

Krish6488 11 months, 2 weeks ago

Selected Answer: B

I would go with B as v3-32 TPU offers much more computational power than a single P100 GPU, and this upgrade should provide a substantial decrease in training time.

Also tf.distributestrategy is good to perform distreibuted training on multiple GPUs or TPUs but the current setup has just one GPU which makes it the second best option provided the architecture uses multiple GPUs.

Increase in memory may allow large batch size but wont address the fundamental problem which is over utilised GPU

Early stopping is good for avoiding overfitting when model already starts performing at its best. Its good to reduce overall training time but wont improve the training speed

upvoted 4 times

pico 1 year, 1 month ago

Selected Answer: B

Given the options and the goal of decreasing training time, options B (using TPUs) and D (distributed training) are the most effective ways to achieve this goal

C. Enable early stopping in your Vertex AI Training job:

Early stopping is a technique that can help save training time by monitoring a validation metric and stopping the training process when the metric stops improving. While it can help in terms of stopping unnecessary training runs, it may not provide as substantial a speedup as other options.

upvoted 2 times

tavva_prudhvi 11 months, 2 weeks ago

TPUs (Tensor Processing Units) are Google's custom-developed application-specific integrated circuits (ASICs) used to accelerate machine learning workloads. They are often faster than GPUs for specific types of computations. However, not all models or training pipelines will benefit from TPUs, and they might require code modification to fully utilize the TPU capabilities.

upvoted 1 times

andresvelasco 1 year, 1 month ago

Selected Answer: C

A. Increase the instance memory to 512 GB and increase the batch size.
> this will not necessarily decrease training time
B. Replace the NVIDIA P100 GPU with a v3-32 TPU in the training job. Most Voted
> TPU can sacrifice performance
C. Enable early stopping in your Vertex AI Training job.
> YES, this decreases training time without sacrificing performance, if set properly
D. Use the tf.distribute.Strategy API and run a distributed training job.

> No idea .... But I believe the type of machine and architecture cannot be changed as per the wording of the question.

upvoted 1 times

tavva_prudhvi 11 months, 2 weeks ago

Early stopping is a method that allows you to stop training once the model performance stops improving on a validation dataset. While it can prevent overfitting and save time by stopping unnecessary training epochs, it does not inherently speed up the training process.

upvoted 1 times

PST21 1 year, 2 months ago

Option D, using the tf.distribute.Strategy API for distributed training, can be beneficial for improving training efficiency, but it would require additional resources and complexity to set up compared to simply using a TPU.

Therefore, replacing the NVIDIA P100 GPU with a v3-32 TPU in the Vertex AI Training job would be the most effective way to decrease training time while maintaining or even improving model performance

upvoted 2 times

[Removed] 1 year, 3 months ago

Selected Answer: B

I don't understand why so many people are voting for D (tf.distribute.Strategy API). If we look at our training infrastructure, we can see the bottleneck is obviously the GPU, which has 12GB or 16GB memory depending on the model (https://www.leadtek.com/eng/products/ai_hpc(37)/tesla_p100(761)/detail). This means we can afford to have a batch size of only 6-8 images (2GB each) even if we assume the GPU is utilized 100%. And remember the training size is 3M, which means each epoch will have 375-500K steps in the best case.

With 32-cores and 128GB memory, we are able to afford higher batch sizes (e.g., 32), so moving to TPU will accelerate the training.

A is wrong because we can't afford a larger batch size with the current GPU. D is wrong because you don't have multiple GPUs and your current GPU is saturated. C is a viable option, but it seems less optimal than B.

upvoted 4 times

[Removed] 1 year, 3 months ago

I should note that the batch size should be lower than even 6-8 images because the model weights will also take the GPU memory.

upvoted 1 times

Load full discussion...

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.