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 130 DISCUSSI...

Actual exam question from Google's Professional Machine Learning Engineer

Question #: 130

Topic #: 1

[All Professional Machine Learning Engineer Questions]

You are working on a system log anomaly detection model for a cybersecurity organization. You have developed the model using TensorFlow, and you plan to use it for real-time prediction. You need to create a Dataflow pipeline to ingest data via Pub/Sub and write the results to BigQuery. You want to minimize the serving latency as much as possible. What should you do?

A. Containerize the model prediction logic in Cloud Run, which is invoked by Dataflow.

B. Load the model directly into the Dataflow job as a dependency, and use it for prediction.

C. Deploy the model to a Vertex AI endpoint, and invoke this endpoint in the Dataflow job.

D. Deploy the model in a TFServing container on Google Kubernetes Engine, and invoke it in the Dataflow job.

Show Suggested Answer

by pshemol at *Dec. 21, 2022, 1:15 p.m.*

Comments

Type your comment...

Submit

⊟ **guilhermebutzke** Highly Voted 9 months, 1 week ago

Selected Answer: C

C. According Google:
"Instead of deploying the model to an endpoint, you can use the RunInference API to serve machine learning models in your Apache Beam pipeline. This approach has several advantages, including flexibility and portability. However, deploying the model in Vertex AI offers many additional benefits, such as the platform's built-in tools for model monitoring, TensorBoard, and model registry governance.
Vertex AI also provides the ability to use Optimized TensorFlow runtime in your endpoints. To do this, simply specify the TensorFlow runtime container when you deploy your model."

https://cloud.google.com/blog/products/ai-machine-learning/streaming-prediction-with-dataflow-and-vertex

upvoted 5 times

SausageMuffins Most Recent 6 months ago

Selected Answer: B

It's a toss up between B and C.

I chose B because using vertex AI as an endpoint introduces network latency which naturally does not meet the criteria of "minimizing latency".

However, choosing option B also implies that I have more overhead by directly running the model in the dataflow pipeline. Since the question didn't mention any limitations on resources, I assumed that the resources can be scaled accordingly to minimize latency. I might be overthinking on this option though seeing how most of Google questions have a strong preference on their "recommended platforms" like vertex AI. Most of the questions and the community answers seem to tend towards anything that mentions "vertex ai".

upvoted 2 times

guilhermebutzke 9 months, 1 week ago

According Google:
"Instead of deploying the model to an endpoint, you can use the RunInference API to serve machine learning models in your Apache Beam pipeline. This approach has several advantages, including flexibility and portability. However, deploying the model in Vertex AI offers many additional benefits, such as the platform's built-in tools for model monitoring, TensorBoard, and model registry governance.
Vertex AI also provides the ability to use Optimized TensorFlow runtime in your endpoints. To do this, simply specify the TensorFlow runtime container when you deploy your model."

https://cloud.google.com/blog/products/ai-machine-learning/streaming-prediction-with-dataflow-and-vertex

upvoted 2 times

tavva_prudhvi 1 year, 4 months ago

Selected Answer: C

In this case, the best way to minimize the serving latency of the system log anomaly detection model is to deploy it to a Vertex AI endpoint. This will allow Dataflow to invoke the model directly, without having to load it into the job as a dependency. This will significantly reduce the serving latency, as Dataflow will not have to wait for the model to load before it can make a prediction.

Option B would involve loading the model directly into the Dataflow job as a dependency. This would also add an additional layer of latency, as Dataflow would have to load the model into memory before it could make a prediction.

upvoted 3 times

Voyager2 1 year, 5 months ago

C. Deploy the model to a Vertex AI endpoint, and invoke this endpoint in the Dataflow job
https://cloud.google.com/architecture/minimizing-predictive-serving-latency-in-machine-learning

upvoted 1 times

julliet 1 year, 5 months ago

Selected Answer: C

C
I eliminate B because Dataflow is a batch-prediction solution, not real-time

upvoted 1 times

> **7cb0ab3** 7 months ago
>
> Dataflow has a streaming pipeline solution as well.
>
> upvoted 1 times

M25 1 year, 6 months ago

Selected Answer: C

Went with C

upvoted 1 times

Antmal 1 year, 6 months ago

Selected Answer: C

I believe it is C when deploying the model to a Vertex AI endpoint it provides a dedicated prediction service optimised for real-time inference. Vertex AI endpoints are designed for high performance and low latency, making them ideal for real-time prediction use cases. Dataflow can easily invoke the Vertex AI endpoint to perform predictions, minimising serving latency.

upvoted 1 times

hghdh5454 1 year, 7 months ago

Selected Answer: B

B. Load the model directly into the Dataflow job as a dependency, and use it for prediction.

By loading the model directly into the Dataflow job as a dependency, you minimize the serving latency since the model is available within the pipeline itself. This way, you avoid additional network latency that would be introduced by invoking external services, such as Cloud Run, Vertex AI endpoints, or TFServing containers.

upvoted 4 times

Antmal 1 year, 6 months ago

Actually in retrospect C is the correct answer, not B because loading the model directly into the Dataflow job as a dependency may cause unnecessary overhead, as Dataflow jobs are primarily designed for batch processing and may not be optimized for real-time prediction. Additionally, loading the model as a dependency may increase the size of the Dataflow job and introduce complexity in managing dependencies.

upvoted 1 times

wlts 1 year, 7 months ago

Selected Answer: B

By loading the model directly into the Dataflow job as a dependency, you can perform predictions within the same job. This approach helps minimize serving latency since there is no need to make external calls to another service or endpoint. Instead, the model is directly available within the Dataflow pipeline, allowing for efficient and fast processing of the streaming data.

upvoted 1 times

TNT87 1 year, 8 months ago

Selected Answer: C

C. Deploy the model to a Vertex AI endpoint, and invoke this endpoint in the Dataflow job.

The reason for this choice is that deploying the model to a Vertex AI endpoint and invoking it in the Dataflow job is the most efficient and scalable option for real-time prediction. Vertex AI provides a fully managed, serverless platform for deploying and serving machine learning models. It allows for high availability and low-latency serving of models, and can handle a large volume of requests in parallel. Invoking the model via an endpoint in the Dataflow job minimizes the latency for model prediction, as it avoids any unnecessary data transfers or containerization

upvoted 2 times

TNT87 1 year, 6 months ago

Using private endpoints to serve online predictions with Vertex AI provides a low-latency, secure connection to the Vertex AI online prediction service. This guide shows how to configure private endpoints on Vertex AI by using VPC Network Peering to peer your network with the Vertex AI online prediction service

https://cloud.google.com/vertex-ai/docs/predictions/using-private-endpoints

Answer C

upvoted 1 times

shankalman717 1 year, 8 months ago

Selected Answer: C

Option B, loading the model directly into the Dataflow job as a dependency and using it for prediction, may not provide the optimal performance because Dataflow may not be optimized for low-latency predictions.

upvoted 1 times

John_Pongthorn 1 year, 8 months ago

These are anwser
https://cloud.google.com/dataflow/docs/notebooks/run_inference_tensorflow

https://beam.apache.org/documentation/sdks/python-machine-learning/

https://beam.apache.org/documentation/transforms/python/elementwise/runinference/

upvoted 2 times

John_Pongthorn 1 year, 9 months ago

C OR B.
it is straightforward that it should be C as the follwing link https://cloud.google.com/architecture/detecting-anomalies-in-financial-transactions

financial transactions
but B it seems a newer way. it keeps questionable.

upvoted 1 times

Yajnas_arpohc 1 year, 7 months ago

Reading through this link, look like dataflow itself is doing prediction directly .. so B

upvoted 1 times

ares81 1 year, 10 months ago

Selected Answer: C

C, for me.

upvoted 1 times

hiromi 1 year, 10 months ago

Selected Answer: C

C
- https://cloud.google.com/architecture/detecting-anomalies-in-financial-transactions
- https://cloud.google.com/blog/topics/financial-services/detect-anomalies-in-real-time-forex-data-with-ml

upvoted 4 times

pshemol 1 year, 10 months ago

Selected Answer: B

https://cloud.google.com/blog/products/data-analytics/influsing-ml-models-into-production-pipelines-with-dataflow

upvoted 2 times

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.