DUMPSARENA

Professional Machine Learning Engineer

Google Professional-Machine-Learning-Engineer

Version Demo

Total Demo Questions: 7

Total Premium Questions: 145

Buy Premium PDF

https://dumpsarena.com

sales@dumpsarena.com

QUESTION NO: 1

You have a functioning end-to-end ML pipeline that involves tuning the hyperparameters of your ML model using AI Platform, and then using the best-tuned parameters for training. Hypertuning is taking longer than expected and is delaying the downstream processes. You want to speed up the tuning job without significantly compromising its effectiveness. Which actions should you take?

Choose 2 answers

A. Decrease the number of parallel trials

B. Decrease the range of floating-point values

C. Set the early stopping parameter to TRUE

D. Change the search algorithm from Bayesian search to random search.

E. Decrease the maximum number of trials during subsequent training phases.

ANSWER: C E

Explanation:

Reference: https://cloud.google.com/ai-platform/training/docs/hyperparameter-tuning-overview

https://cloud.google.com/ai-platform/training/docs/using-hyperparameter-tuning#early-stopping

QUESTION NO: 2

You are training a Resnet model on AI Platform using TPUs to visually categorize types of defects in automobile engines. You capture the training profile using the Cloud TPU profiler plugin and observe that it is highly input-bound. You want to reduce the bottleneck and speed up your model training process. Which modifications should you make to the tf .data dataset?

Choose 2 answers

A. Use the interleave option for reading data

B. Reduce the value of the repeat parameter

C. Increase the buffer size for the shuffle option.

D. Set the prefetch option equal to the training batch size

E. Decrease the batch size argument in your transformation

ANSWER: D E

Explanation:

https://towardsdatascience.com/overcoming-data-preprocessing-bottlenecks-with-tensorflow-data-service-nvidia-dali-and-other-d6321917f851

QUESTION NO: 3

Your team needs to build a model that predicts whether images contain a driver's license, passport, or credit card. The data engineering team already built the pipeline and generated a dataset composed of 10,000 images with driver's licenses, 1,000 images with passports, and 1,000 images with credit cards. You now have to train a model with the following label map: ['driversjicense', 'passport', 'credit_card']. Which loss function should you use?

A. Categorical hinge

B. Binary cross-entropy

C. Categorical cross-entropy

D. Sparse categorical cross-entropy

ANSWER: C

Explanation:

- **Categorical entropy** is better to use when you want to **prevent the model from giving more importance to a certain class**. Or if the **classes are very unbalanced** you will get a better result by using Categorical entropy.

- But **Sparse Categorical Entropy** is a more optimal coice if you have a huge amount of classes, enough to make a lot of memory usage, so since sparse categorical entropy uses less columns it **uses less memory**.

https://stats.stackexchange.com/questions/326065/cross-entropy-vs-sparse-cross-entropy-when-to-use-one-over-the-other

QUESTION NO: 4

You are a data scientist at an industrial equipment manufacturing company. You are developing a regression model to estimate the power consumption in the company's manufacturing plants based on sensor data collected from all of the plants. The sensors collect tens of millions of records every day. You need to schedule daily training runs for your model that use all the data collected up to the current date. You want your model to scale smoothly and require minimal development work. What should you do?

A. Develop a custom TensorFlow regression model, and optimize it using Vertex AI Training.

B. Develop a regression model using BigQuery ML.

C. Develop a custom scikit-learn regression model, and optimize it using Vertex AI Training

D. Develop a custom PyTorch regression model, and optimize it using Vertex AI Training

ANSWER: B

Explanation:

BigQuery ML is a powerful tool that allows you to build and deploy machine learning models directly within BigQuery, Google's fully-managed, serverless data warehouse. It allows you to create regression models using SQL, which is a familiar

and easy-to-use language for many data scientists. It also allows you to scale smoothly and require minimal development work since you don't have to worry about cluster management and it's fully-managed by Google.

BigQuery ML also allows you to run your training on the same data where it's stored, this will minimize data movement, and thus minimize cost and time.

References:

QUESTION NO: 5

You work for a toy manufacturer that has been experiencing a large increase in demand. You need to build an ML model to reduce the amount of time spent by quality control inspectors checking for product defects. Faster defect detection is a priority. The factory does not have reliable Wi-Fi. Your company wants to implement the new ML model as soon as possible. Which model should you use?

A. AutoML Vision model

B. AutoML Vision Edge mobile-versatile-1 model

C. AutoML Vision Edge mobile-low-latency-1 model

D. AutoML Vision Edge mobile-high-accuracy-1 model

ANSWER: A

QUESTION NO: 6

You are the Director of Data Science at a large company, and your Data Science team has recently begun using the Kubeflow Pipelines SDK to orchestrate their training pipelines. Your team is struggling to integrate their custom Python code into the Kubeflow Pipelines SDK. How should you instruct them to proceed in order to quickly integrate their code with the Kubeflow Pipelines SDK?

A. Use the func_to_container_op function to create custom components from the Python code.

B. Use the predefined components available in the Kubeflow Pipelines SDK to access Dataproc, and run the custom code there.

C. Package the custom Python code into Docker containers, and use the load_component_from_file function to import the containers into the pipeline.

D. Deploy the custom Python code to Cloud Functions, and use Kubeflow Pipelines to trigger the Cloud Function.

ANSWER: D

QUESTION NO: 7

You have trained a DNN regressor with TensorFlow to predict housing prices using a set of predictive features. Your default precision is tf.float64, and you use a standard TensorFlow estimator;

estimator = tf.estimator.DNNRegressor(

feature_columns=[YOUR_LIST_OF_FEATURES],

hidden_units-[1024, 512, 256],

dropout=None)

Your model performs well, but Just before deploying it to production, you discover that your current serving latency is 10ms @ 90 percentile and you currently serve on CPUs. Your production requirements expect a model latency of 8ms @ 90 percentile. You are willing to accept a small decrease in performance in order to reach the latency requirement Therefore your plan is to improve latency while evaluating how much the model's prediction decreases. What should you first try to quickly lower the serving latency?

A. Increase the dropout rate to 0.8 in_PREDICT mode by adjusting the TensorFlow Serving parameters

B. Increase the dropout rate to 0.8 and retrain your model.

C. Switch from CPU to GPU serving

D. Apply quantization to your SavedModel by reducing the floating point precision to tf.float16.

ANSWER: D

Explanation:

Applying quantization to your SavedModel by reducing the floating point precision can help reduce the serving latency by decreasing the amount of memory and computation required to make a prediction. TensorFlow provides tools such as the tf.quantization module that can be used to quantize models and reduce their precision, which can significantly reduce serving latency without a significant decrease in model performance.

Reference: https://www.tensorflow.org/guide/quantization