

# Assignment 7: optional catchup assignment 2 - VERTEX AI - for midterm and quiz - this will catch up midterm.

## a) Vertex AI Workbench:

Build an image classification model with transfer learning and the notebook executor

### Objective:

How to configure and launch notebook executions with Vertex AI Workbench.

- Use parameters in a notebook
- Configure and launch notebook executions from the Vertex AI Workbench UI

In this lab, you'll use transfer learning to train an image classification model on the [DeepWeeds dataset](#) from [TensorFlow Datasets](#). You'll use [TensorFlow Hub](#) to experiment with feature vectors extracted from different model architectures, such as [ResNet50](#), [Inception](#), and [MobileNet](#), all pre-trained on the [ImageNet benchmark dataset](#). By leveraging the notebook executor via the Vertex AI Workbench UI, you'll launch jobs on Vertex AI Training that use these pre-trained models and retrain the last layer to recognize the classes from the DeepWeeds dataset.

## Set up Cloud environment

### Step 1: Enable the Compute Engine API

console.cloud.google.com/compute/instances?project=cmpe260-334300&authuser=2&orgonly=true&supportedpurview=organizationId

LEARN MORE UPGRADE

Google Cloud Platform CMPE260 VM instances CREATE INSTANCE IMPORT VM REFRESH START / RESUME STOP SUSPEND RESET DELETE CREATE SCHEDULE OPERATIONS HELP ASSISTANT SHOW INFO PANEL LEARN

**Compute Engine**

Virtual machines VM instances INSTANCE SCHEDULE

VM instances are highly configurable virtual machines for running workloads on Google infrastructure. Learn more

Filter Enter property name or value

| Status | Name                           | Zone          | Recommendations | In use by | Internal IP      | External IP    | Connect |
|--------|--------------------------------|---------------|-----------------|-----------|------------------|----------------|---------|
| Green  | tensorflow-2-3-20211209-130458 | us-central1-a |                 |           | 10.128.0.4 (red) | 35.233.119.126 | SSH     |
| Green  | tensorflow-2-7-20211208-091022 | us-central1-a |                 |           | 10.128.0.3 (red) | None           | SSH     |

Related actions

- View billing report
- Monitor VMs
- Explore VM logs
- Set up firewall rules
- Patch management

DISMISSED

## Step 2: Enable the Vertex AI API

console.cloud.google.com/vertex-ai?referrer=search&authuser=2&orgonly=true&project=cmpe260-334300&supportedpurview=organizationId

LEARN MORE UPGRADE

Google Cloud Platform CMPE260 Vertex AI Dashboard

**Get started with Vertex AI**

Vertex AI empowers machine learning developers, data scientists, and data engineers to take their projects from ideation to deployment, quickly and cost-effectively. [Learn more](#)

Try an interactive tutorial to learn how to train, evaluate, and deploy a Vertex AI AutoML or custom-trained model [VIEW TUTORIALS](#)

Region us-central1 (Iowa)

Recent datasets

- iowa\_daily 23 minutes ago
- + CREATE DATASET

Recent models

- beans-model-pipeline 1 hour ago
- beans-model-pipeline 1 hour ago
- + TRAIN NEW MODEL

Get predictions

After you train a model, you can use it to get predictions, either online as an endpoint or through batch requests

Recent notebook instances

- tensorflow-2-3-20211209-130458 58 minutes ago
- tensorflow-2-7-20211208-091022 3 hours ago
- + CREATE NOTEBOOK INSTANCE

Recent endpoints

ONLINE TRAFFIC REQUESTS ERROR RATE

No data is available for the selected time frame

## Step 3: Create a Vertex AI Workbench instance

The screenshot shows the Google Cloud Platform Vertex AI Workbench interface. On the left, there's a sidebar with options like Dashboard, Datasets, Features, Labeling tasks, and Workbench (which is selected). Under Workbench, there are sub-options: Pipelines, Training, Experiments, Models, Endpoints, Batch predictions, and Metadata. The main area is titled "Notebooks" and shows a table of managed notebooks. The columns include Notebook name, Zone, Auto-upgrade, Environment, Machine type, GPUs, Permission, and Last modified. There are two entries: "tensorflow-2-3-20211209-130458" (Zone: us-central1-a, Environment: TensorFlow 2.3, Machine type: 4 vCPUs, 15 GB RAM, GPU: None, Permission: Service account, Last modified: Dec 9, 2021, 2:34:45 PM) and "tensorflow-2-7-20211208-091022" (Zone: us-central1-a, Environment: TensorFlow 2.7, Machine type: 4 vCPUs, 15 GB RAM, GPU: None, Permission: Service account, Last modified: Dec 9, 2021, 11:57:12 AM).

## Create Managed Notebook:

This screenshot shows the "Create a managed notebook" dialog box. It has several sections: 
 - \*\*Basic Configuration\*\*: A dropdown for "None" and a note about GPU types and disk size. It includes fields for "Data disk type" (Standard Persistent Disk) and "Data disk size in GB" (set to 100).
 - \*\*Disk Encryption\*\*: Options for "Google-managed encryption key" (selected) and "Customer-managed encryption key (CMK)".
 - \*\*Idle Shutdown\*\*: Options for "Enable Idle Shutdown" and a field for "Time of inactivity before shutdown (Minutes)" set to 60.
 - \*\*Networking\*\*: Options for "Google-managed networks" (selected), "Networks in this project", "Private service access required", and "Networks shared with me".
 - \*\*Security\*\*: Options for "Enable nbconvert" and "Enable file downloading from Notebook UI".
 - \*\*Permission\*\*: A field for "Owner" containing "makeeteersjsu@gmail.com".
 - \*\*Buttons\*\*: "CREATE" and "CANCEL" buttons at the bottom.

This screenshot shows the "Managed notebooks" list. The table has columns: Notebook name, Location, Owner, and Last modified. There are two entries:
 - "london-bikes-codeblob" (Location: us-central1-b, Owner: ragurs@cisco.com, Last modified: Nov 30, 2021, 12:01:14 AM)
 - "managed-notebook-1639094160" (Location: us-central1-b, Owner: ragurs@cisco.com, Last modified: Dec 9, 2021, 3:57:35 PM)

## Write training code

# Step 1: Download and preprocess dataset

The screenshot shows a Jupyter Notebook interface with the following code:

```
Import the necessary libraries
[1]: import tensorflow as tf
import tensorflow_datasets as tfds
import tensorflow_hub as hub

Download the data from TensorFlow Datasets and extract the number of classes and dataset size.
[2]: data, info = tfds.load(name='deep_weeds', as_supervised=True, with_info=True)
NUM_CLASSES = info.features['label'].num_classes
DATASET_SIZE = info.splits['train'].num_examples

Downloading and preparing dataset 469.32 MiB (download: 469.32 MiB, generated: 469.99 MiB, total: 939.31 MiB) to /home/jupyter/tensorflow_datasets/deep_weeds/3.0.0...
DL Completed... 100% [2/2 [00:16<00:00, 4.83s/uri]
DL Size... 100% [468/468 [00:16<00:00, 57.06 MiB/s]

Extraction completed... 100% [1/1 [00:16<00:00, 16.13s/file]

Dataset deep_weeds downloaded and prepared to /home/jupyter/tensorflow_datasets/deep_weeds/3.0.0. Subsequent calls will reuse this data.
2021-12-10 00:37:00.335897: W tensorflow/stream_executor/platform/default/dso_loader.cc:64] Could not load dynamic library 'libcudnn.so.1'; dlsym or dlopen error: libcudnn.so.1: cannot open shared object file: No such file or directory; LD_LIBRARY_PATH: /usr/local/cuda/lib64:/usr/local/cuda/lib:/usr/local/lib/x86_64-linux-gnu:/usr/local/nvidia/lib64:/usr/local/nvidia/lib:/usr/local/nvidia/lib64
2021-12-10 00:37:00.337027: W tensorflow/stream_executor/cuda/cuda_driver.cc:269] Failed call to cuInit: UNKNOWN ERROR (383)
2021-12-10 00:37:00.337027: I tensorflow/stream_executor/cuda/cuda_diagnostics.cc:156] kernel driver does not appear to be running on this host (vm-eb37d7a1-6ab0-4ed6-89b0-0e098395f929). Make sure that the driver is loaded. Try a reboot.
2021-12-10 00:37:00.357850: I tensorflow/core/platform/cpu_feature_guard.cc:151] This TensorFlow binary is optimized with oneAPI Deep Neural Network Library (oneDNN) to use the following CPU instructions in performance-critical operations: AVX2 FMA
To enable them in other operations, rebuild TensorFlow with the appropriate compiler flags.
```

# Step 2: Create model

The screenshot shows a Jupyter Notebook interface with the following code:

```
[3]: def preprocess_data(image, label):
    image = tf.image.resize(image, (300,300))
    return tf.cast(image, tf.float32) / 255., label

[4]: # Create train/validation splits
# Shuffle dataset
dataset = data['train'].shuffle(1000)

train_split = 0.8
val_split = 0.2
train_size = int(train_split * DATASET_SIZE)
val_size = int(val_split * DATASET_SIZE)

train_data = dataset.take(train_size)
train_data = train_data.map(preprocess_data)
train_data = train_data.batch(64)

validation_data = dataset.skip(train_size)
validation_data = validation_data.map(preprocess_data)
validation_data = validation_data.batch(64)

[5]: feature_extractor_model = "inception_v3"
```

# Model Compile and Fit

```

  val_split = 0.2
  train_size = int(train_split * DATASET_SIZE)
  val_size = int(val_split * DATASET_SIZE)

  train_data = dataset.take(train_size)
  train_data = train_data.map(preprocess_data)
  train_data = train_data.batch(64)

  validation_data = dataset.skip(train_size)
  validation_data = validation_data.map(preprocess_data)
  validation_data = validation_data.batch(64)

[5]: feature_extractor_model = "inception_v3"

[7]: tf_hub_uri = f"https://tfhub.dev/google/imagenet/{feature_extractor_model}/feature_vector/5"

[8]: feature_extractor_layer = hub.KerasLayer(
    tf_hub_uri,
    trainable=False)

[10]: model = tf.keras.Sequential([
    feature_extractor_layer,
    tf.keras.layers.Dense(units=NUM_CLASSES)
])

[*]: model.compile(
    optimizer=tf.keras.optimizers.Adam(),
    loss=tf.keras.losses.SparseCategoricalCrossentropy(from_logits=True),
    metrics=['acc'])

model.fit(train_data, validation_data=validation_data, epochs=20)
Epoch 1/20
114/219 [=====.....] - ETA: 10:53 - loss: 1.2528 - acc: 0.5873

```

## Execute notebook

Import the necessary libraries

```

  [1]: import tensorflow as tf
  import tensorflow_datasets as tfds
  import tensorflow_hub as hub

```

Download the data from TensorFlow Datasets and extract the number of classes and dataset size.

```

  [2]: data, info = tfds.load(name='deep_weeds', as_supervised=True, with_info=True)
  NUM_CLASSES = info.features['label'].num_classes
  DATASET_SIZE = info.splits['train'].num_examples

```

Downloading and preparing dataset 469.32 MiB (download: 469.32 MiB, generated: 469.99 MiB, total: 939.31 MiB) to /home/jupyter

Completed... 100% 2/2 [00:16<00:00, 4.83s/ url]

DL Size... 100% 468/468 [00:16<00:00, 57.06 MiB/s]

Extraction completed... 100% 1/1 [00:16<00:00, 16.13s/file]

Dataset deep\_weeds downloaded and prepared to /home/jupyter/tensorflow\_datasets/deep\_weeds/3.0.0. Subsequent calls will reuse

2021-12-10 00:37:08.333589: W tensorflow/stream\_executor/platform/default/dso\_loader.cc:64] Could not load dynamic library 'libcuda.so.1': /usr/local/cuda/lib64/libcuda.so.1: cannot open shared object file: No such file or directory@/lib/x86\_64-linux-gnu/libcudnn.so.8.0.0@@HEDT

2021-12-10 00:37:08.333782: W tensorflow/stream\_executor/cuda/cuda\_driver.cc:269] failed call to cuInit: UNKNOWN ERROR (303)

2021-12-10 00:37:08.333793: I tensorflow/stream\_executor/cuda/cuda\_diagnostics.cc:156] kernel driver does not appear to be running: /proc/driver/nvidia/version does not exist

2021-12-10 00:37:08.357850: I tensorflow/core/platform/cpu\_feature\_guard.cc:151] This TensorFlow binary is optimized with one wing CPU instructions in performance-critical operations: AVX2 FMA

To enable them in other operations, rebuild TensorFlow with the appropriate compiler flags.

Define a preprocessing function to scale the image data by 255.

```

  [3]: def preprocess_data(image, label):
        image = tf.image.resize(image, (300,300))
        return tf.cast(image, tf.float32) / 255., label

```

Submit notebooks to Executor

Notebook: Untitled.ipynb

Execution name: image\_example\_1639097809753

Cloud Storage bucket: tg\_model\_test

Machine type: 4 CPUs, 16 GB RAM

Accelerator type: NVIDIA Tesla P4

Type: One-time execution

Environment: TensorFlow Enterprise 2.6 (GPU)

Input parameters (optional): feature\_extractor\_model=resnet\_v2\_50

SUBMIT CANCEL

← → ⌂ console.cloud.google.com/vertex-ai/workbench/list/executions?authuser=5&orgonly=true&project=legalpa-sandbox-4o9d&supportedpurview=organizationId

Apps SOA Governance Reinforcement Le... Java Design Patte... mail258/machinele... Midterm Numeric... Machine Learning Design Patterns The Java(TM) We... Java Interview Qu... FAOSTAT Java Interview qu... Reading List

Google Cloud Platform LegalPA

Notebooks REFRESH DELETE vertex

MANAGED NOTEBOOKS PREVIEW USER-MANAGED NOTEBOOKS EXECUTIONS PREVIEW SCHEDULED PREVIEW

This page shows your notebook executions created through Notebook Executor. You can share execution results with others by clicking the icon in the table below.

Region: us-central1 (Iowa)

| Execution name              | Source Notebook                       | Execution time          | Schedule |
|-----------------------------|---------------------------------------|-------------------------|----------|
| image_example_1639097809753 | untitled_at_2021_12_09_17_10_17.ipynb | Dec 9, 2021, 5:10:17 PM | None     |

← → ⌂ console.cloud.google.com/vertex-ai/locations/us-central1/training/3860377628540141568?authuser=5&orgonly=true&project=legalpa-sandbox-4o9d&supportedpurview=organizationId

Apps SOA Governance Reinforcement Le... Java Design Patte... mail258/machinele... Midterm Numeric... Machine Learning Design Patterns The Java(TM) We... Java Interview Qu... FAOSTAT Java Interview qu... Reading List

Google Cloud Platform LegalPA

image\_example\_1639097809753

Training began at Dec 9, 2021, 5:10:18 PM and is still in progress.

|                                    |  |
|------------------------------------|--|
| Status                             | Pending  |
| Custom job ID                      | 3860377628540141568  |
| Created                            | Dec 9, 2021, 5:10:18 PM  |
| Start time                         | Dec 9, 2021, 5:10:18 PM  |
| Elapsed time                       | 1 min 29 sec   |
| Region                             | us-central1  |
| Encryption type                    | Google-managed key   |
| Machine type (Worker pool 0)       | n1-standard-4  |
| Machine count (Worker pool 0)      | 1  |
| Accelerator (Worker pool 0)        | NVIDIA_TESLA_P4  |
| Accelerator count (Worker pool 0)  | 1  |
| Container Location (Worker pool 0) | gcr.io/deeplearning-platform-release/tf2-gpu-2-6-latest  |
| Arguments (Worker pool 0)          | nbexecutor; --input-notebook; gs://tg_model_test/executor_files/image_example_1639097809753/Untitled.ipynb; --output-notebook; gs://tg_model_test/executor_files/image_example_1639097809753/untitled_at_2021_12_09_17_10_17.ipynb; -parameters; feature_extractor_model=resnet_v2_50; -kernel_name; python3 |
| Dataset                            | No managed dataset   |
| Algorithm                          | Custom training  |
| Objective                          | Custom   |
| Container (Training)               | Custom   |
| Logs                               | <a href="#">View logs</a>  |

[VIEW CUSTOM JOB INPUTS IN JSON](#)

← → ⌂ console.cloud.google.com/vertex-ai/locations/us-central1/training/3860377628540141568/cpu?authuser=5&orgonly=true&project=legalpa-sandbox-4o9d&supportedpurview=organizationId

Apps SOA Governance Reinforcement Le... Java Design Patte... mail258/machinele... Midterm Numeric... Machine Learning Design Patterns The Java(TM) We... Java Interview Qu... FAOSTAT Java Interview qu... Reading List

Google Cloud Platform LegalPA

image\_example\_1639097809753

Custom job was completed on Dec 9, 2021, 5:58:46 PM.

|                                    |  |
|------------------------------------|--|
| Status                             | Finished   |
| Custom job ID                      | 3860377628540141568  |
| Created                            | Dec 9, 2021, 5:10:18 PM  |
| Start time                         | Dec 9, 2021, 5:20:31 PM  |
| Elapsed time                       | 38 min 15 sec  |
| Region                             | us-central1  |
| Encryption type                    | Google-managed key   |
| Machine type (Worker pool 0)       | n1-standard-4  |
| Machine count (Worker pool 0)      | 1  |
| Accelerator (Worker pool 0)        | NVIDIA_TESLA_P4  |
| Accelerator count (Worker pool 0)  | 1  |
| Container Location (Worker pool 0) | gcr.io/deeplearning-platform-release/tf2-gpu-2-6-latest  |
| Arguments (Worker pool 0)          | nbexecutor; --input-notebook; gs://tg_model_test/executor_files/image_example_1639097809753/Untitled.ipynb; --output-notebook; gs://tg_model_test/executor_files/image_example_1639097809753/untitled_at_2021_12_09_17_10_17.ipynb; -parameters; feature_extractor_model=resnet_v2_50; -kernel_name; python3 |
| Dataset                            | No managed dataset   |
| Algorithm                          | Custom training  |
| Objective                          | Custom   |
| Container (Training)               | Custom   |
| Logs                               | <a href="#">View logs</a>  |

[VIEW CUSTOM JOB INPUTS IN JSON](#)

CPU GPU NETWORK

**CPU utilization**

Percent

COLLAPSE ALL EXPAND ALL

This screenshot shows the Google Cloud Platform Notebooks interface. The top navigation bar includes links for Apps, SOA Governance, Reinforcement Le..., Java Design Patte..., mail258/machinele..., Midterm Numeric..., Machine Learning, Design Patterns, The Java(TM) We..., Java Interview Qu..., FAOSTAT, Java interview qu..., and a Reading List. The main title is "Google Cloud Platform" with a "vertex" search bar. Below the title, there are tabs for "Notebooks" (selected), "REFRESH", and "DELETE". Under "Notebooks", there are sub-tabs for "MANAGED NOTEBOOKS" (selected), "USER-MANAGED NOTEBOOKS", "EXECUTIONS" (selected), and "SCHEDULES". A search bar and filter icon are also present. The main content area displays a table of notebook executions. The table has columns for "Execution name", "Source Notebook", "Execution time", and "Schedule". One execution is listed: "image\_example\_1639097809753" from "united\_st\_2021\_12\_09\_17\_10\_17.ipynb" at "Dec 9, 2021, 5:10:17 PM" with "None" scheduled. A "VIEW RESULT" button is next to the execution name.

## Clean Up

### Delete the Execution and notebook

This screenshot shows the Google Cloud Platform Notebooks interface, similar to the previous one but with a different view. The main content area displays a table with the header "Execution name", "Source Notebook", "Execution time", and "Schedule". A message below the table states "No executions to display".

This screenshot shows the Google Cloud Platform Managed notebooks interface. The top navigation bar and tabs are identical to the previous screenshots. The main content area displays a table of managed notebooks. The table has columns for "Notebook name", "Location", "Owner", and "Last modified". Two notebooks are listed: "london-bikes-codelab" and "managed-notebook-1639094160". The "managed-notebook-1639094160" row is selected. A modal dialog box titled "Delete notebook" is open in the foreground, asking "Are you sure you want to delete notebook 'managed-notebook-1639094160'? (This will also delete boot disk 'managed-notebook-1639094160')". It contains "CANCEL" and "DELETE" buttons.

