Build, Train, Save, Deploy and Test a Convolutional Neural Network Model using MNIST

Introduction

This lab will use the [MNIST](#) computer vision data set to train a deep learning model to recognize handwritten digits. A simple convolutional neural network built using Keras will be submitted to Watson Machine Learning for training. The trained model will be saved in the model repository, deployed, and scored. The lab will use a notebook to programmatically accomplish these tasks.

Objectives

Upon completing the lab, you will know how to programmatically:

1. Set up the data and result buckets in IBM Cloud Storage
2. Set up the training definition
3. Train the model
4. Monitor the training progress and results
5. Save and Deploy the Trained Model
6. Test the Deployed Model

Lab Steps

Step 1 - Create a Jupyter Notebook

1. Click on the hamburger icon ☰, then click on **Projects**, and then **Watson Studio Labs** (or whatever you named the project)
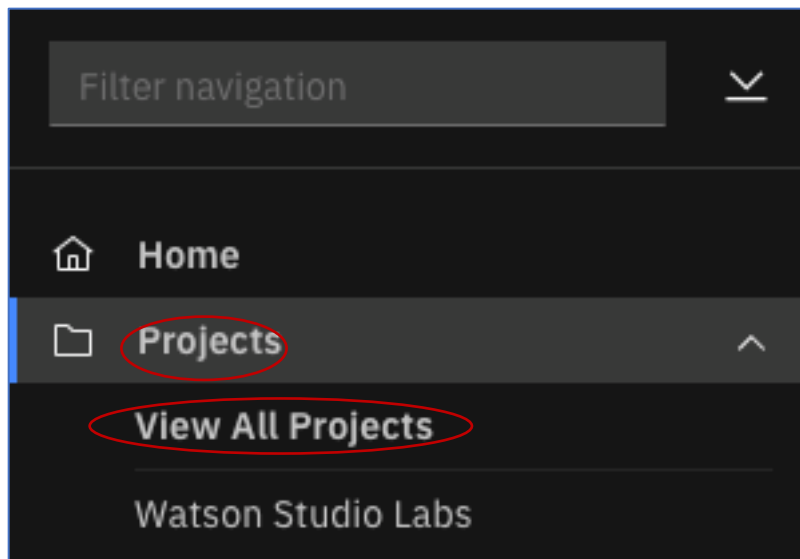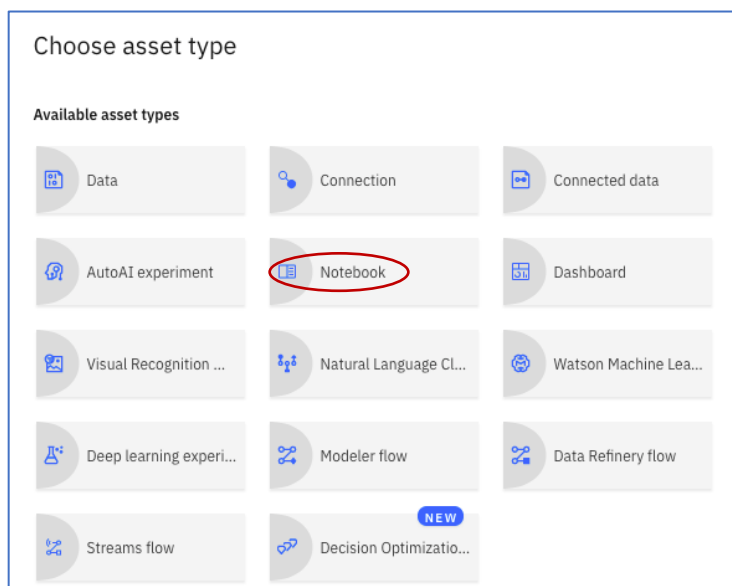
2. We are now going to create a notebook in our project. This notebook will be created from a url that points to the Keras-MNIST example notebook in the github repository. Click the **Add to project** link.
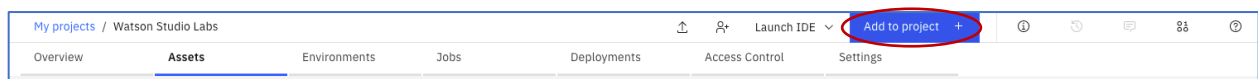
3. Click on **Notebook**

4. Click on **From URL** under **New Notebook,** enter **MNIST-Example** for the **Name**, and optionally enter a **Description**. Leave the default for the **Runtime**.

   User Copy Hyperlink and paste the following url into the **Notebook URL** field.

https://github.com/bleonardb3/ML_POT_10-29-2020/blob/main/Lab-7/MNIST-Example.ipynb

Click **Create Notebook.**

5. Before executing the code in the notebook, we need to do the following:
   a. Obtain the Cloud Object Storage credentials.
   b. Obtain an api key and the location of the WatsonMachineLearning service. We need to copy this information into a Notebook cell. This information is required to work with the Watson Machine Learning API. Note, we already did this in Lab-5. Go to the Lab-5 notebook to get these credentials.

6. Right-click on **IBM Cloud Pak for Data**, and the click on **Open Link in New Tab**.
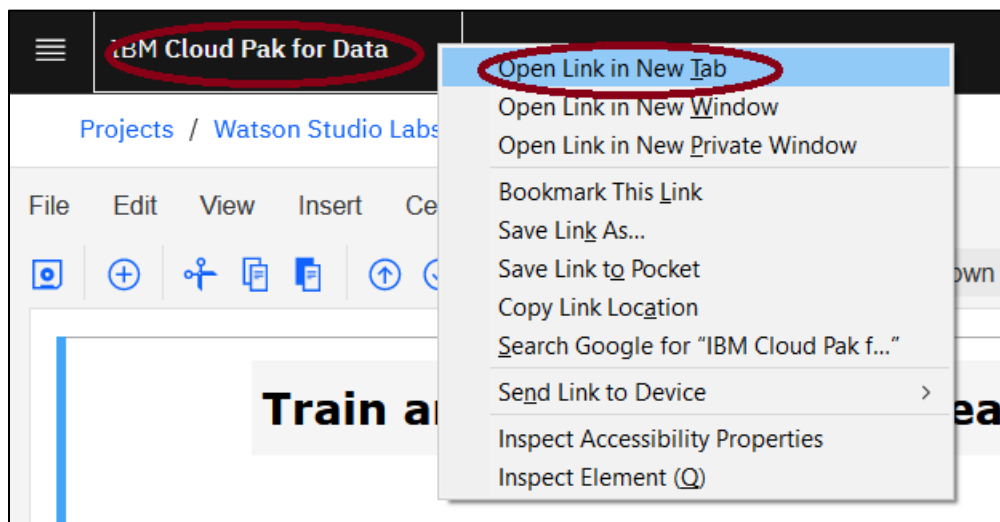
7. Click on the new **IBM Cloud Pak for Data** browser tab.

8. Click on the hamburger icon ![hamburger], and click on **Services**, and then **Service instances**.
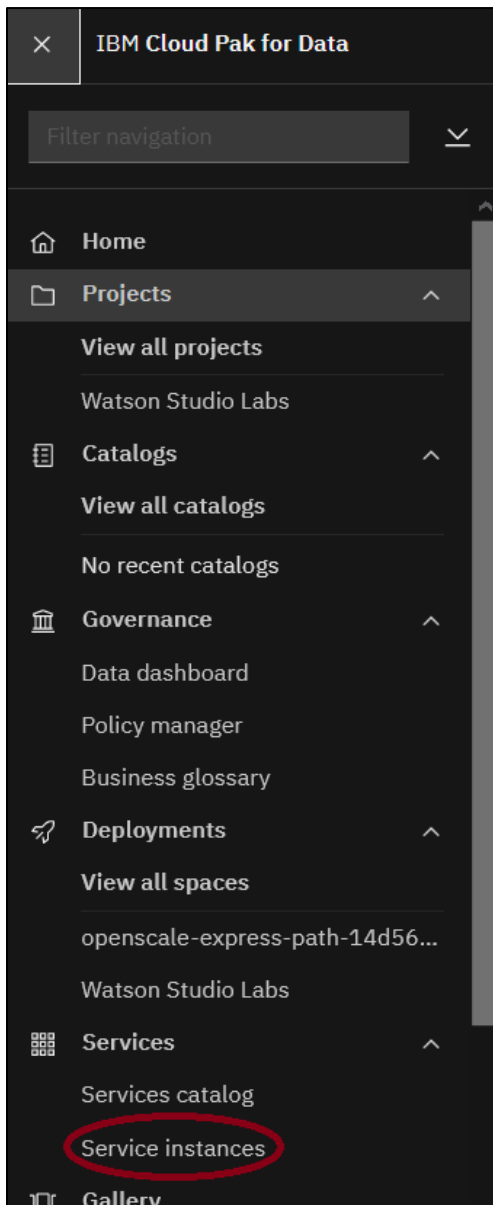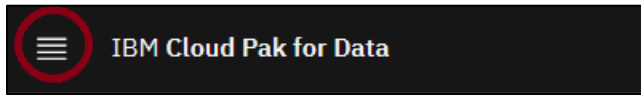
9. Click on vertical ellipse ⋮ on the right side of the Cloud Object Storage service. Click on the **Manage in IBM Cloud** entry.

10. Click on **Service credentials**

11. Click on **New Credentials**.

12. Type in **keras** for the Name and click on the down arrow icon ⌄ adjacent to **Advanced options**.

13. Enable the **Include HMAC Credentials.** This is required for Watson Machine Learning interface to Cloud Object Storage. Click on **Add**.
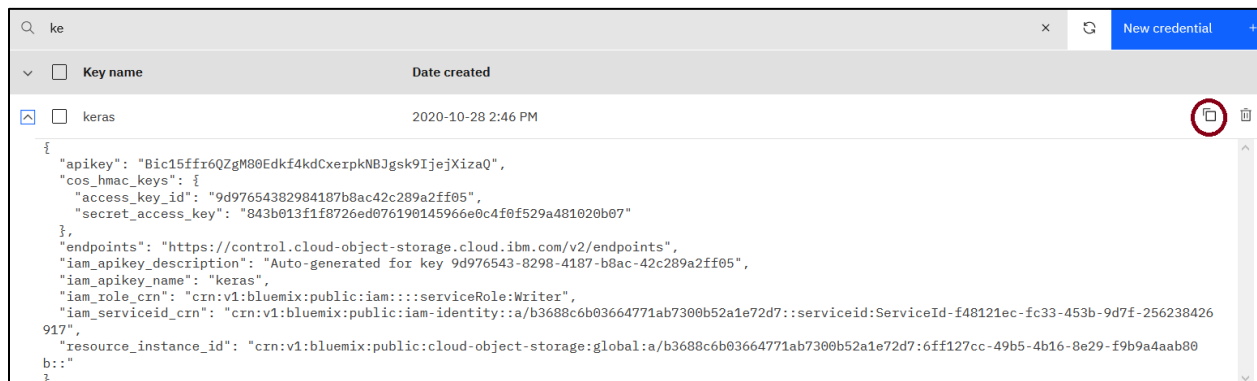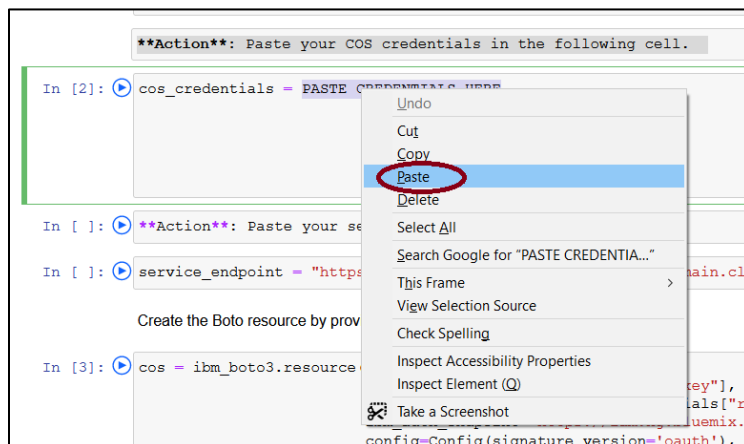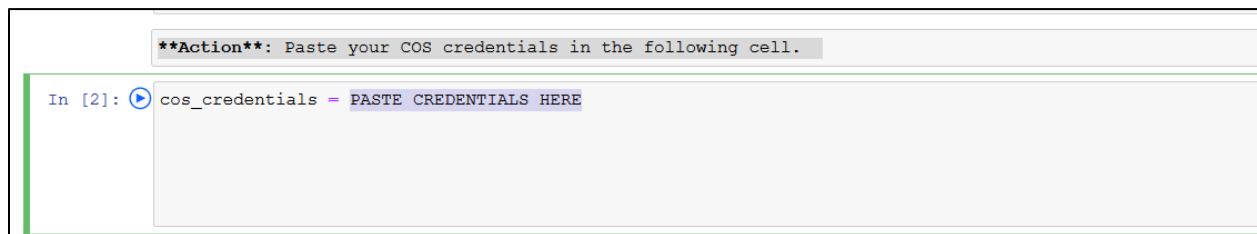
14. Enter keras in the Search field. Click on the down arrow icon ⌄   adjacent to keras.

15. Click on the copy icon ⬜ .

{
  "apikey": "Bic15ffr6QZgM80Edkf4kdCxerpkNBJgsk9IjejXizaQ",
  "cos_hmac_keys": {
    "access_key_id": "9d97654382984187b8ac42c289a2ff05",
    "secret_access_key": "843b013f1f8726ed076190145966e0c4f0f529a481020b07"
  },
  "endpoints": "https://control.cloud-object-storage.cloud.ibm.com/v2/endpoints",
  "iam_apikey_description": "Auto-generated for key 9d976543-8298-4187-b8ac-42c289a2ff05",
  "iam_apikey_name": "keras",
  "iam_role_crn": "crn:v1:bluemix:public:iam:::::serviceRole:Writer",
  "iam_serviceid_crn": "crn:v1:bluemix:public:iam-identity::a/b3688c6b03664771ab7300b52a1e72d7::serviceid:ServiceId-f48121ec-fc33-453b-9d7f-256238426917",
  "resource_instance_id": "crn:v1:bluemix:public:cloud-object-storage:global:a/b3688c6b03664771ab7300b52a1e72d7:6ff127cc-49b5-4b16-8e29-f9b9a4aab80b::"
}

16. Highlight "PASTE YOUR CREDENTIALS HERE" (under the **Action**: Paste your COB credentials in the following cell). Right-click and then click on **Paste.**

17. The credentials should appear similar to below with different values.

18. Return to the Cloud Object Storage tab and click on **Endpoints**.

19. Click on the appropriate geo where you account has been set up.

20. My account is in us-geo. Copy the Public endpoint corresponding to your geo.

21. Return to the notebook, and highlight "PASTE SERVICE ENDPOINT HERE" (underneath the Action: Paste your service endpoint). Right-click and click on **Paste**.
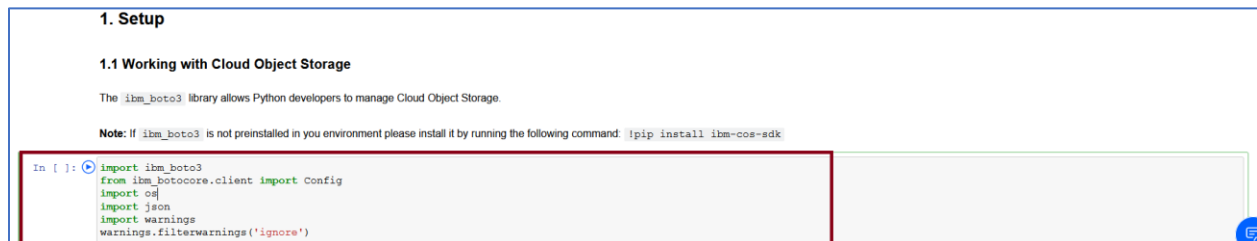
22. The service endpoint should appear similar to below depending on the geo.

Action: Paste your service endpoint

In [6]: ▶ service_endpoint = "https://s3.us.cloud-object-storage.appdomain.cloud"

23. Return to the top of the notebook, read through the documentation in the beginning and then select the first code cell to execute.

1. Setup

1.1 Working with Cloud Object Storage

The `ibm_boto3` library allows Python developers to manage Cloud Object Storage.

Note: If `ibm_boto3` is not preinstalled in you environment please install it by running the following command: `!pip install ibm-cos-sdk`

```
In [ ]: ▶ import ibm_boto3
         from ibm_botocore.client import Config
         import os
         import json
         import warnings
         warnings.filterwarnings('ignore')
```

For those not familiar with Jupyter notebooks, read below.

A Jupyter notebook consists of a series of cells. These cells are of 2 types (1) documentation cells containing markdown, and (2) code cells (denoted by a bracket on the left of the cell) where you write Python code, R, or Scala code depending on the type of notebook. Code cells can be run by putting the cursor in the code cell and pressing **<Shift><Enter>** on the keyboard. Alternatively, you can execute the cells by clicking on **Run icon** on the menu bar that will run the current cell (where the cursor is located) and then select the cell below. In this way, repeatedly clicking on **Run** executes all the cells in the notebook. When a code cell is executed the brackets on the left change to an asterisk '*' to indicate the code cell is executing. When completed, a sequence number appears. The output, if any, is displayed below the code cell.

Execute each of the notebook cells in order (either by typing in **<Shift><Enter>** or using the **Run** menu option). Read the notebook documentation to gain an understanding of the code that is executing.

24. Click Watson Studio Labs to exit out of the notebook.

IBM **Cloud Pak for Data**

Projects / Watson Studio Labs / Heart Disease
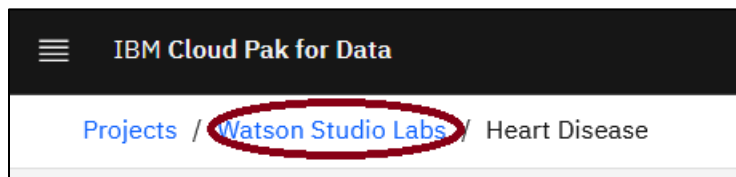
You have completed the Lab!

- ✓ Set up the data and result buckets in IBM Cloud Storage
- ✓ Set up the training definition
- ✓ Trained the model
- ✓ Monitored the training progress and results
- ✓ Saved and Deployed the Trained Model
- ✓ Tested the Deployed Model