**IBM Cloud**

A Hands-on Introduction to

Deep Learning and Neural Network

Modeling with IBM Watson Studio

Training and Deploying a deep learning model for digit handwriting recognition

**Lab Guide**

Notices and Disclaimers

© Copyright IBM Corporation 2018.

The information contained in these materials is provided for informational purposes only, and is provided AS IS without warranty of any kind, express or implied. IBM shall not be responsible for any damages arising out of the use of, or otherwise related to, these materials. Nothing contained in these materials is intended to, nor shall have the effect of, creating any warranties or representations from IBM or its suppliers or licensors, or altering the terms and conditions of the applicable license agreement governing the use of IBM software. References in these materials to IBM products, programs, or services do not imply that they will be available in all countries in which IBM operates. This information is based on current IBM product plans and strategy, which are subject to change by IBM without notice. Product release dates and/or capabilities referenced in these materials may change at any time at IBM’s sole discretion based on market opportunities or other factors, and are not intended to be a commitment to future product or feature availability in any way.

This document is current as of the initial date of publication and may be changed by IBM at any time. Not all offerings are available in every country in which IBM operates.

IBM, the IBM logo and ibm.com are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both. If these and other IBM trademarked terms are marked on their first occurrence in this information with a trademark symbol (® or ™), these symbols indicate U.S. registered or common law trademarks owned by IBM at the time this information was published. Such trademarks may also be registered or common law trademarks in other countries. A current list of IBM trademarks is available on the Web at “Copyright and trademark information” at ibm.com/legal/copytrade.shtml

Other company, product and service names may be trademarks or service marks of others

**Table of Contents**

[Lab Environment Overview 4](#_Toc515004196)

[Module 1: Create an IBM Cloud Account and Watson Studio Project- Lab Instructions 5](#_Toc515004197)

[Module 2: Create a Model Flow to Design a Neural Network for MNIST Data 12](#_Toc515004198)

[Module 3: Building and Deploying a TensorFlow Model 25](#_Toc515004199)

[Conclusion 32](#_Toc515004200)

# Lab Environment Overview

|  |  |
| --- | --- |
| Purpose: | The purpose of this lab is to allow you to explore how you can develop and deploy deep learning models in Watson Studio. Specifically, this includes,   * Working with a visual representation of a Neural Network example in the Model Flow editor * Training and optimizing your model using the Experiment Builder * Deploying the model to IBM Watson Machine Learning where it can be utilized by developers for use in applications   For this lab, we will employ a standard MNIST dataset which is a sample of handwritten digits that will be used for the training and analysis or our model.  By the end of this lab, you will be familiar with the following:   * Navigating IBM Watson Studio * Creating and deploying Deep Learning Neural Network models in Watson Studio * Familiarity with IBM Cloud Services |
|  |  |
| Tasks: | Tasks you will complete in this lab exercise include:   * Set up your IBM Account with Watson Studio * Creating a model for the MNIST data and running an experiment * Refinement and deployment of an existing TensorFlow model, using hyperparameter optimizations |

# Module 1: Create an IBM Cloud Account and Watson Studio Project- Lab Instructions

| Step | Action |
| --- | --- |
| 1 | **Set up your IBM Cloud Account with Watson Studio**   1. Log into IBM Data Science Experience at <http://dataplatform.ibm.com/> 2. Under IBM Watson Studio, click “Learn More”. When the Watson Studio page appears, click “Start your Free Trial”. 3. Follow the instructions to create an account or use an existing IBM Cloud account.        1. You will need to log into your email account to confirm and complete account registration. 2. Select the “Confirm Account” link in the email to be redirected to IBM Cloud.        1. When you first sign into your new account, you will need to select and organization and space. 2. We will be using the default options for this lab. Click “Continue”.      1. Once you see that the Object-Storage is done provisioning, click “Get Started”. |
| 2 | **Create your Watson Studio Project**   1. Click the “New Project” icon.      1. For project type, select “Complete”      1. In order to create a project, you will need to have a place to store your project’s assets. Watson Studio projects utilize an associated IBM Cloud Object Storage (COS) services for this purpose. There is a free tier of COS that can be easily created. To do this, click “Add” under “Select Storage Service” on the new project page.      1. Select “Lite” for the service type and Click “Create”      1. Click “Confirm”.      1. Once back in the new project page, click “Refresh”, enter “Digit Analysis” for the project name, and click “Create”.      1. You have now successfully created your Watson Studio project. You should see your project’s main page. |

# Module 2: Create a Model Flow to Design a Neural Network for MNIST Data

In this module, you will use the Watson Studio Model Flow editor and Experiment Builder to design and build a single layer Neural Network model for the MNIST dataset.

| Step | Action |
| --- | --- |
| 1 | **Create a Connection to the Cloud Object Storage**  In order for your project to use the you COS instance as a data source, you need to create a connect to the service. For this, you will need the COS credentials you noted in earlier.   1. In your Watson Studio project, select the “Add to Project” menu and select “Connection”.      1. In the “IBM Services” section, select “Cloud Object Storage” (not infrastructure).      1. Name your connection “COS Connection” and complete the form with the credentials found at the following location (without the quotes). Please note that “Access Key” and “Secret Key” are not required. When complete, click “Create”.   <http://bit.ly/2kCaypD>     1. When complete, you will be brought to your project’s asset overview page, with the connection you just created listed under “Data Assets”. |
| 2 | **Create a Model Flow**   1. From your project’s overview page, click the “Add to Project” menu and select “Modeler Flow”     In Modeler Flow creation page, select “From Example”. Select the “Single Convolution layer on MNIST” and click “Create”.     1. You will be brought to the model canvas that has been populated with our example model.      1. In order to finalize this, we first need to point it to the data files the model will use for training and validation. Perform the following: 2. Double-click the first node, labeled “Image Data” (this opens the node details side panel.) 3. In the details side panel, in the “DATA” section, in the “Data Connections” drop-down menu, select the Cloud Object Storage connection that you created in the previous step. 4. In the “Buckets” drop-down menu, select “dl-lab”. 5. In the “Train data file” drop-down menu, select "mnist-train.pkl". 6. In the “Test data file” drop-down menu, select "mnist-test.pkl". 7. In the “Validation data file” drop-down menu, select "mnist-validate.pkl". 8. Click “Save”.      1. Next, you will need to make some changes to the final node (SGD). 2. Double-click the final node, labeled “SGD” (this opens the node details side panel.) 3. Set “Learning rate” to 0.001 4. Set “Decay” to 0 5. Click “Save”.      1. You can now save our neural network design as a training definition. 2. Click the publish icon (  ). 3. Specify a name for the training definition. 4. In the “Specify WML Instance” drop-down menu, select the Watson Machine Learning service instance that is associated with the project. 5. Click the “Publish” button. |
| 3 | **Train your Model using the Experiment Builder**   1. From your project’s overview page, click the “Add to Project” menu and select “Experiment”      1. Complete the following in the new Experiment page:    1. Name your experiment “MNIST Experiment 1”    2. In the “Machine Learning Service” drop-down, select the Watson Machine Learning instance associated with your project    3. In the area for “Cloud Object Storage” click “Select”.    4. In the “Cloud Object Storage connection” drop-down list, select the Cloud Object Storage connection you created before.    5. For the “Training Data Bucket”, select the training data bucket you created before.    6. For the “Training Results Bucket”, select the results bucket you created before.    7. Click “Create”        1. Add the training definition you created in the previous step.    1. Click **Add training definition**.    2. Click the **Existing training definitions** tab.    3. From the drop-down menu, select the training definition you published from the flow editor.    4. Select "1/2 x NVIDIA Tesla K80 (1 GPU)" for the compute plan.    5. Select "none" for the hyperparameter optimization method.    6. Click “Select”      1. Click **Create and run**      1. You can monitor the status of your training run.      1. Once the training run is completed, you can review the details of the training run results and other output by clicking on the training definition.      1. We will save the model, so it can be deployed into a Watson Machine Learning for testing and use by applications. To do, return to the Experiment page. Select the “Actions” menu for the training definition located in the “Completed” section of the page. From that menu, select “Save Model”. Save the model with the name “MNIST Model 1” and click “Save”. |
| 4 | **Deploy and Test your Model**   1. We will now deploy the model into your Watson Machine Learning Service. Once the model is deployed, you can test it or have developers access it from their applications in production using a simple Rest URL. To deploy the model, return to the “Assets” page of your project. In the “Models” section, you should see your saved model listed. Click on the model.      1. When the model opens, go to the “Deployments” tab and click “Add Deployment”      1. On the create deployment page, select the “Web Service” tab, name your deployment “MNIST Deployment 1” and click “Save”.      1. Wait until you receive an indication that the deployment was successful.      1. Once deployed, we can either test the deployment or provide a URL to developers for use in applications. Click on the deployment name to view the details. Next, click on the “Implementation” tab. Notice the scoring URL, as well as the sample code for accessing it.      1. Click on the “Test” tab. Copy and paste the contents of the “**tf-mnist-test-payload-pkl.json” file you previously downloaded into the “Enter Input Data” field.** 2. Click “Predict”      1. The JSON file contains an encoded image representing the number “7”. Based on this, we would expect our model to correctly identify the image as a “7”. The output of our team will be as shown below. In the output, each field represents a digit (0-9). The “0” or “1” is a true or false indication as to whether or not the image is a specific digit. Notice that the 8th entry is labeled as “1” meaning that the prediction is that our image is a “7”. |

# Module 3: Building and Deploying a TensorFlow Model

In this module, you will use the Experiment Builder to train and deploy an existing TensorFlow model. In addition, you will use hyperparameter optimizations as part of the training process. These optimizations can be used to refine the performance of your model.

| Step | Action |
| --- | --- |
| 1 | **Download Sample Data and Model**   1. Download the 4 MNIST sample data files from the following URL:   <http://yann.lecun.com/exdb/mnist/>   1. Download the example TensorFlow sample code from the following URL   **<Insert URL Here>** |
| 2 | **Create and Run a New Experiment**   1. Go to the overview page of your project. Click on the “Add to Project” menu and select “Experiment”.      1. In the new experiment page, name your experiment “MNIST Experiment TF”. 2. We will need to provide a location for the experiment to access training data and to store results. We use Cloud Object Storage for this purpose. Click on “Select” under the Cloud Object Storage Bucket area.      1. In the Cloud Object Storage page, select “New Connection” to create a connection to your project’s default COS instance. 2. From the list, select your existing COS instance. 3. Click on “New” for both your training and results bucket selection. 4. For you training data bucket name, enter “tf-data” and for training results enter “tf-results” 5. Click “Create”      1. Once the experiment is created, click on the link to the “source” bucket, as shown below. This will open the COS bucket so you can add the training data.      1. Once the bucket is open, click on the “click here to add objects” link and select the MNIST files you downloaded earlier. The files will then be uploaded to the bucket.      1. We can now add our TensorFlow training definition code to our experiment. Return to the experiment page and click “Add Training Definition”. Enter “TensorFlow Experiment with HPO” as the name. 2. Click “browse” and select the training definition file you previously downloaded. 3. Select “TensorFlow 1.5” as the framework      1. For “Compute Plan”, select “1/2 x NVIDIA Tesla K80 (1 GPU)”. 2. This training definition allows for the use of specific hyperparameter optimization (HPO) to enable us to assess the model, based on different conditions and assumptions. When the experiment is run, it can run multiple instances, each with a different HPO values to determine which combination of values provide the best performance and accuracy. For this lab, we will specify value ranges for 4 HPOs. To start, click on the “Hyperparameter optimization method” menu and select “Random”.      1. Additional options will now appear. Set the “number of optimizer steps” field to “4”, “Objective” to “Accuracy” and “Maximize or Minimize” to “maximize”.      1. We will now specify the HPO values that will be used for this experiment. To do this, click “add hyperparameter”.      1. Add the following hyperparameters, as specified in the following table.      1. Click “Create and Run” to run the experiment. The experiment control page will appear. Please note that is may take 5-10 for your experiment to run. |
| 3 | **Review Experiment Results**   1. Once the experiment is complete, select the “Compare Runs” tab.      1. Note here that all four runs had very similar results in this case. Also note that each run used a different set of hyperparameter values, since we asked for a “random” allocation. We can also view the details of a specific run by clicking on its link. 2. Click on the “Training Runs” tab. From here, we can create a model from any of these runs based on the results. Click on the “Actions” menu for the run with the highest accuracy and select “Save Model”. Name the model, “TF Model” and click “Save”. |
| 4 | **Deploy the TensorFlow Model**   1. We can now deploy the model using the same process as in Module 2. To do this, return to the “Assets” page of your project. Under the “Models” section, select the model you just saved. 2. On the model overview page, select the “Deployments” tab. Click on “Add Deployment”.      1. On the deployment page, enter “TF Deployment” as the deployment name and click “Save”. 2. Your model will be deployed when the status changes to “DEPLOY\_SUCCESS” |

# Conclusion

In this lab, we learned how Watson Studio can facilitate the development and deployment of Deep Learning models and applications in a highly collaborative and open environment. We welcome you continue to explore this other capabilities of Watson Studio.

**Thank you for your time today!!**

**The IBM Wolfpack Team**