

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

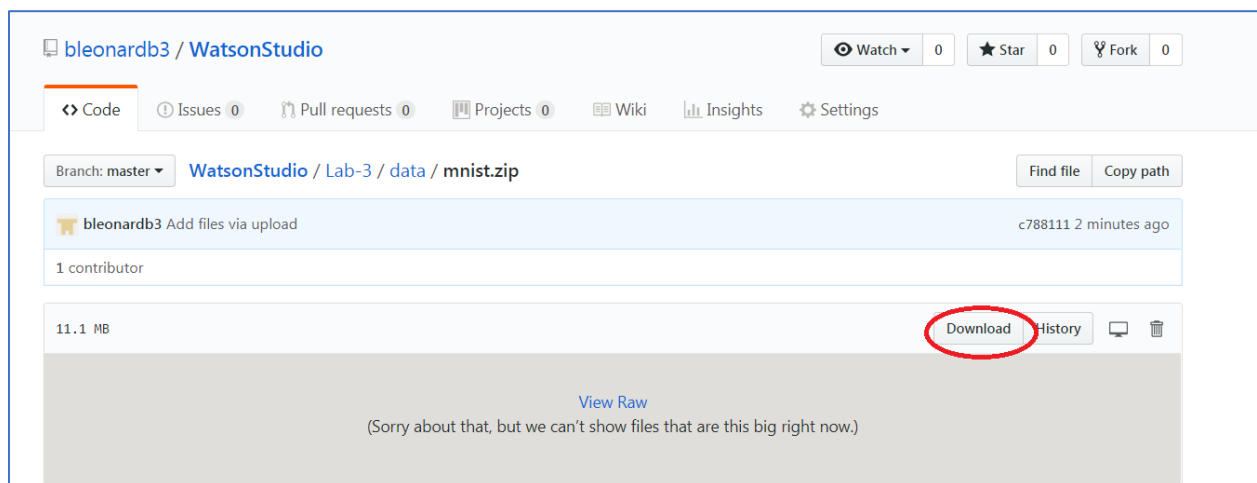
This lab will use the [MNIST](#) computer vision data set to train a deep learning model to recognize handwritten digits. A single layer convolutional neural network will be built in the Watson Studio neural network designer, and then trained using the Watson Studio Experiment Builder. The trained model will be saved in the model repository, deployed, and then tested with sample image data. The lab consists of the following steps:

1. Set up the data files in IBM Cloud Storage.
2. Design the neural network
3. Train the model
4. Monitor the training progress and results
5. Save and Deploy the Trained Model
6. Test the Deployment

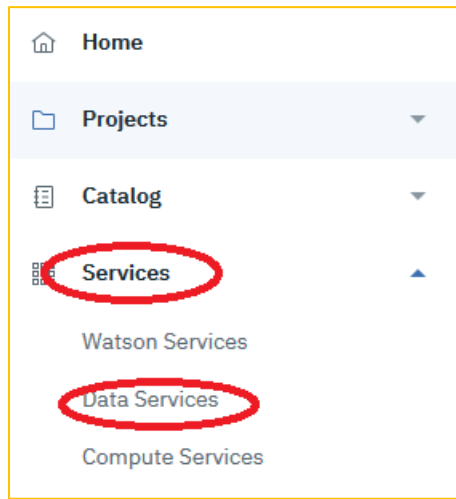
Step 1: Set up the Data Files in IBM Cloud Storage

Training a deep learning model using Watson Machine Learning relies on using Cloud Object Storage for reading input (such as training data) as well as for storing results (such as log files.)

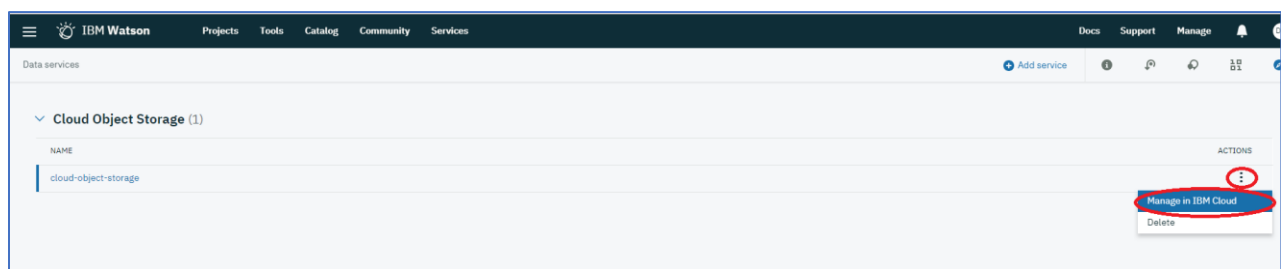
1. Download the [mnist.zip](#) file. Extract the files. Four files should be extracted:
 1. a training file (mnist-tf-train.pkl) – pickle format.
 2. a test file (mnist-tf-test.pkl)
 3. a validation file (mnist-tf-valid.pkl)
 4. test.json (will be used to test the deployed model)



2. Return to Watson Studio, click on the  icon, then click on **Services**, and then **Data Services**.



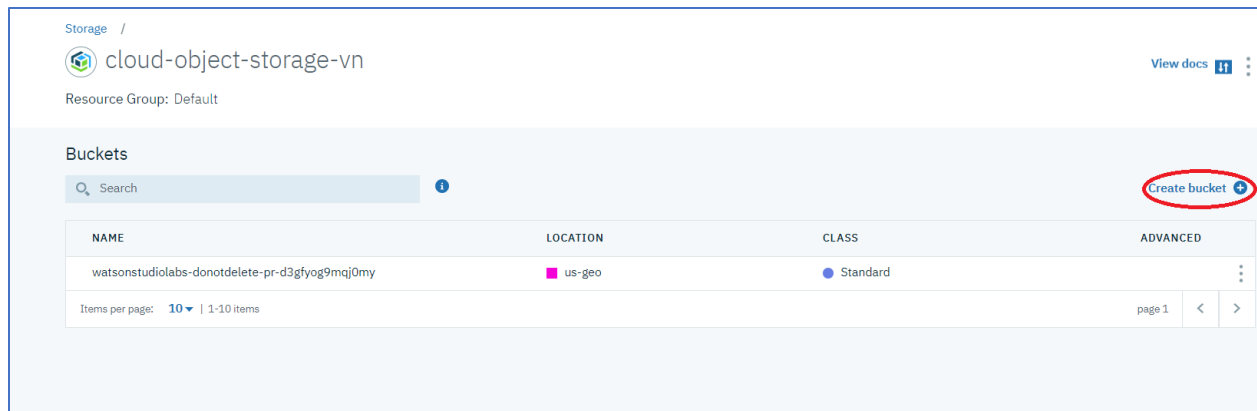
3. Select the vertical **ellipse** on the right-hand side of the cloud object storage entry, and then click on **Manage in IBM Cloud**.



4. A new browser tab **Create Object Storage – IBM Cloud** is created. This is the IBM Cloud user interface to the object storage subsystem



5. Click on **Create bucket**



6. Enter a unique name for the bucket - mnist-lab-train-xxx (replace xxx with your initials), click on **Cross-Region** for the **Resiliency**, and click on **us-geo** for the **Location**. **MAKE SURE YOU CHANGE THE LOCATION TO US-GEO BECAUSE IT DEFAULTS TO AP-GEO**. Scroll down and click on **Create bucket**.

Unique bucket name: [See naming rules](#)

mnist-lab-train-bxb

Resiliency

Location

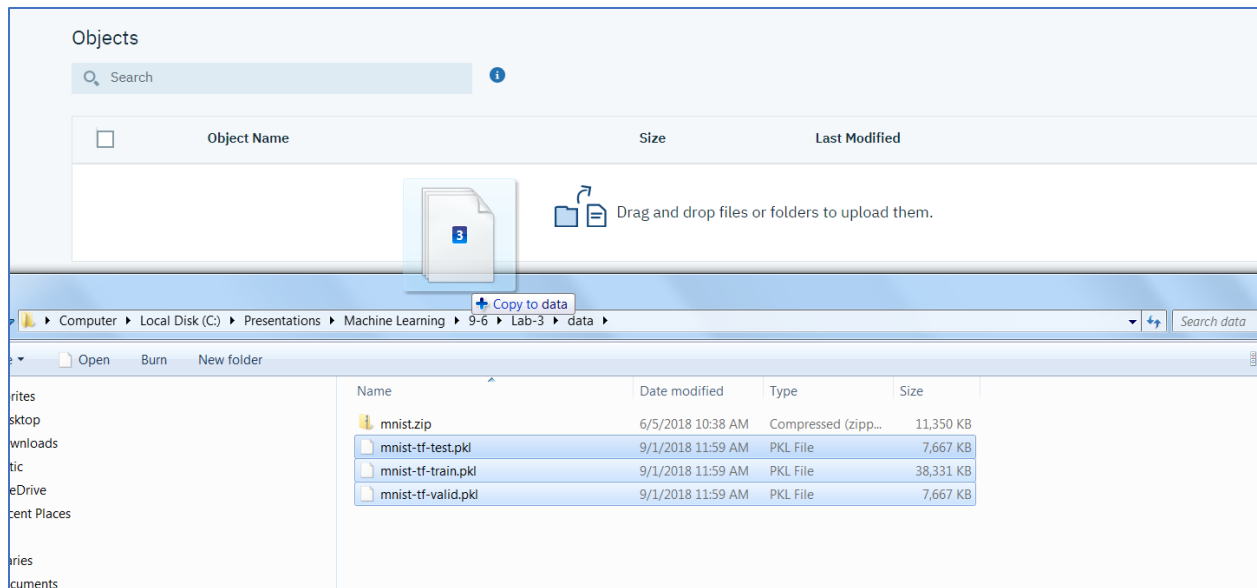
Cross Region

us-geo

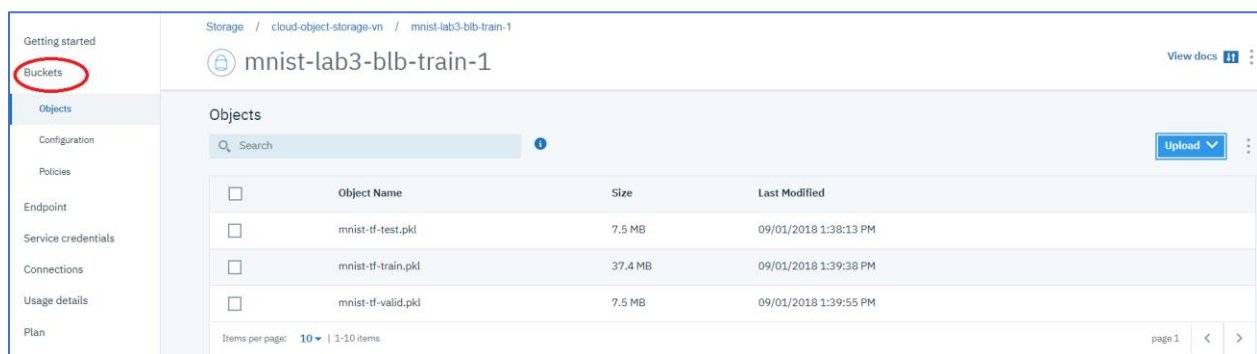
Storage class [See pricing for each class](#)

Standard

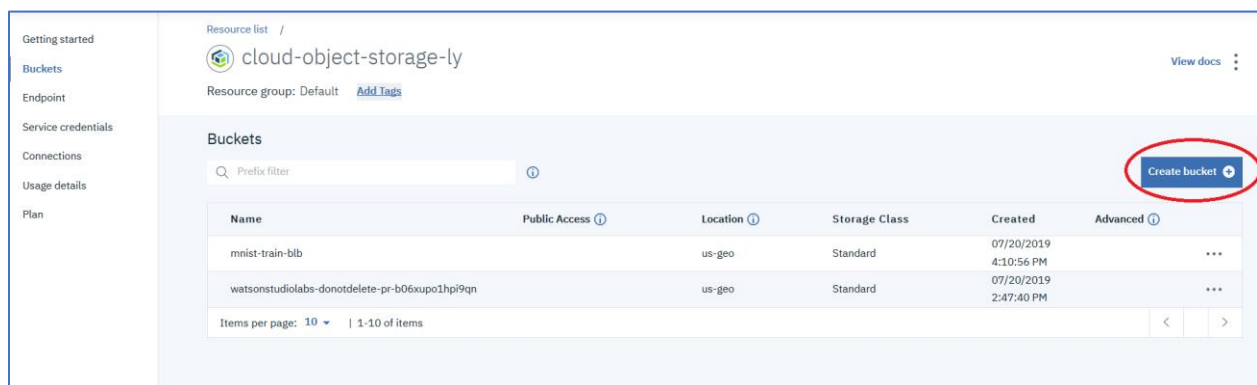
7. Navigate to the directory where the 3 mnist files are stored. Select these 3 files and drag and drop where indicated.



8. Click on **Buckets** to add a second bucket.




9. Click on **Create Bucket**



10. Name it mnist-lab-results-xxx, where xxx are your initials. Follow the procedure above to create the second bucket. No files need to be added. **MAKE SURE YOU CHANGE THE LOCATION TO US-GEO.**

Resource list /

 cloud-object-storage-ly

Resource group: Default [Add Tags](#)

[View docs](#)

Create bucket

Unique bucket name: [See naming rules](#)

mnist-lab-results-blb

Resiliency Location

Cross Region us-geo

Highest availability

Storage class [See pricing for each class](#)

Standard

Additional configuration (optional)


☐ Add Archive rule [?](#)

11. Click on Buckets.

IBM Cloud

Search resources and offerings...

Resource list /

 cloud-object-storage-ly

Resource group: Default [Add Tags](#)

Buckets

12. The Cloud Object Storage panel should appear as below. Note, you may not have the datacatalog bucket listed.

Buckets

Q Prefix filter [?](#)

[Create bucket](#)

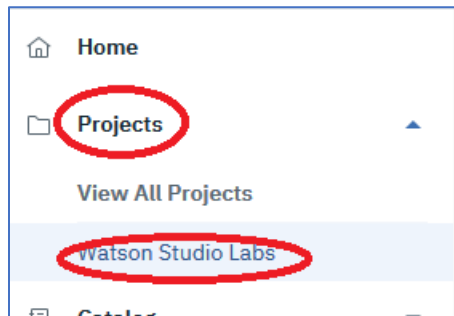
Name	Location ?	Storage Class	Created	Advanced ?
mnist-lab-results-bxb	▲ us-geo	Standard	02/06/2019 1:55:01 PM	...
mnist-lab-train-bxb	▲ us-geo	Standard	02/06/2019 12:52:28 PM	...
watsonstudiolabs-datacatalog-b1nvdome	■ us-south	Standard	01/20/2019 5:33:28 PM	...
watsonstudiolabs-donotdelete-pr-o9cdwmatoq6kn0	▲ us-geo	Standard	01/23/2019 4:26:18 PM	...

Items per page: 10 | 1-10 Items

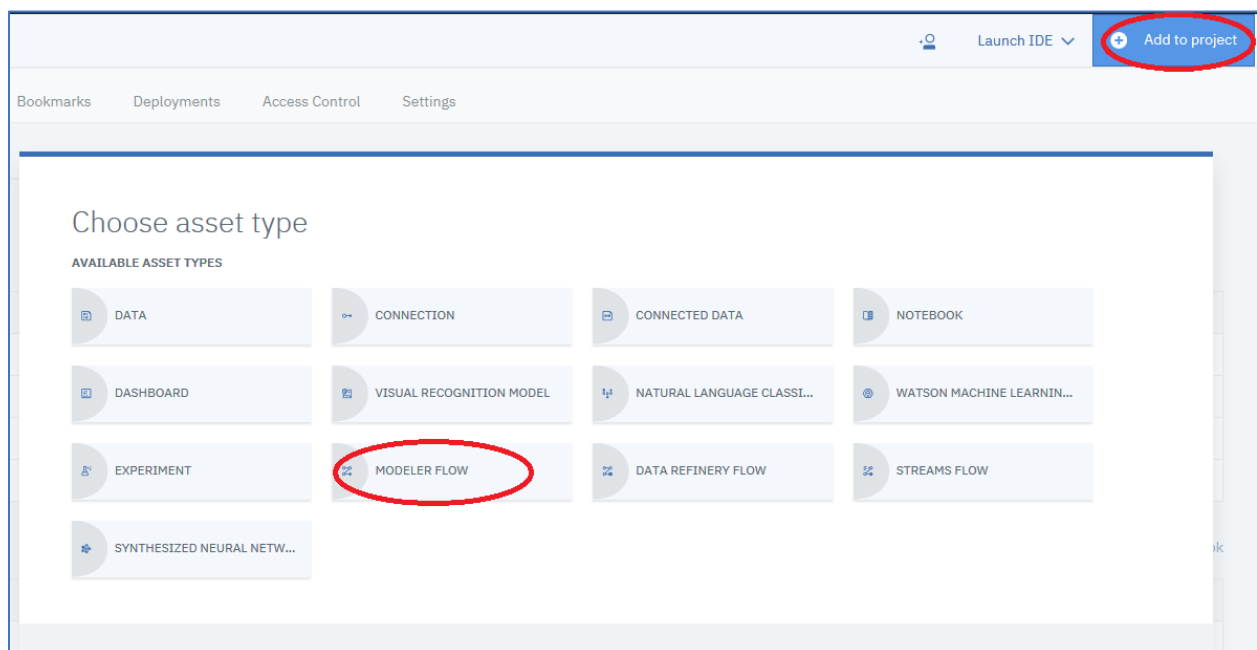
page 1 < >

Step 2: Design the Neural Network and Publish Training Definition

- Return to Watson Studio, and click on the  icon, then click on **Projects**, then **Watson Studio Labs**.



2. Click on the **Add to project** and then click on **MODELER FLOW**.



3. Click on **From example**.

IBM Watson Projects Tools Catalog Community Services

Modeler

New From file **From example**

Name*
Type name here. 50

Description
Type description here. 500

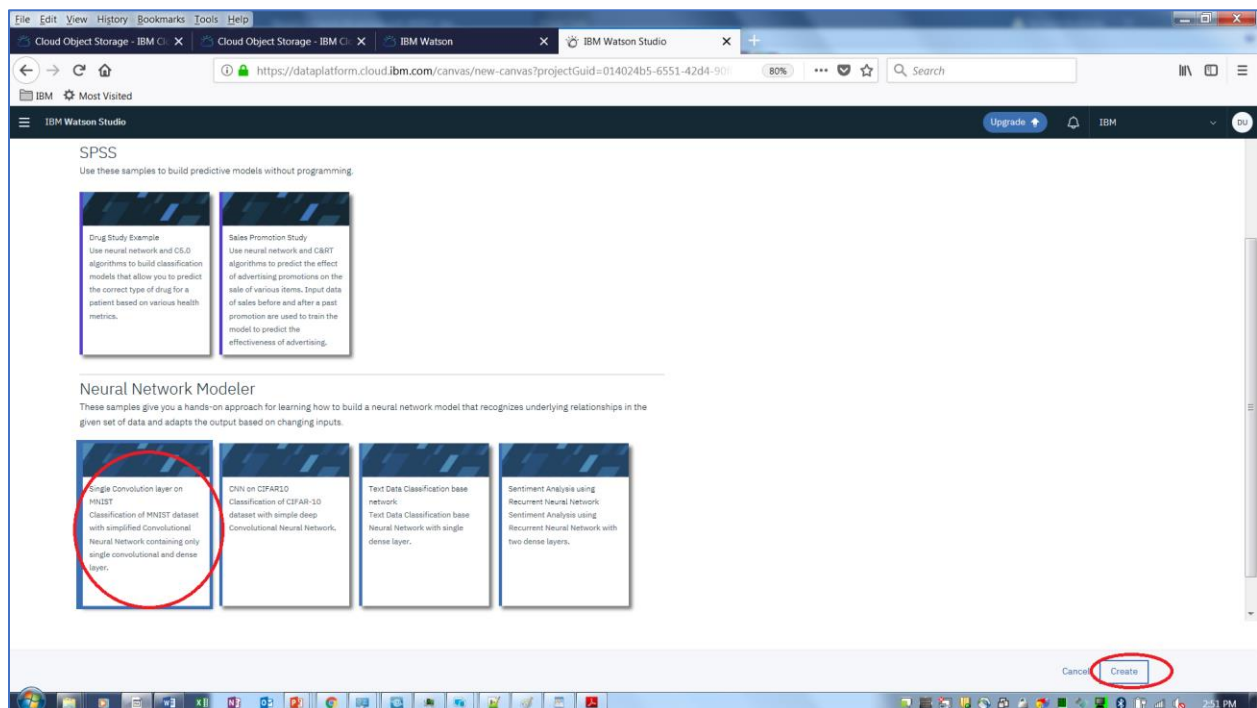
Select flow type

☒ Modeler Flow ☐ Neural Network Modeler BETA

Runtime

☒ IBM SPSS Modeler ☐ Scala Spark 2.1 BETA

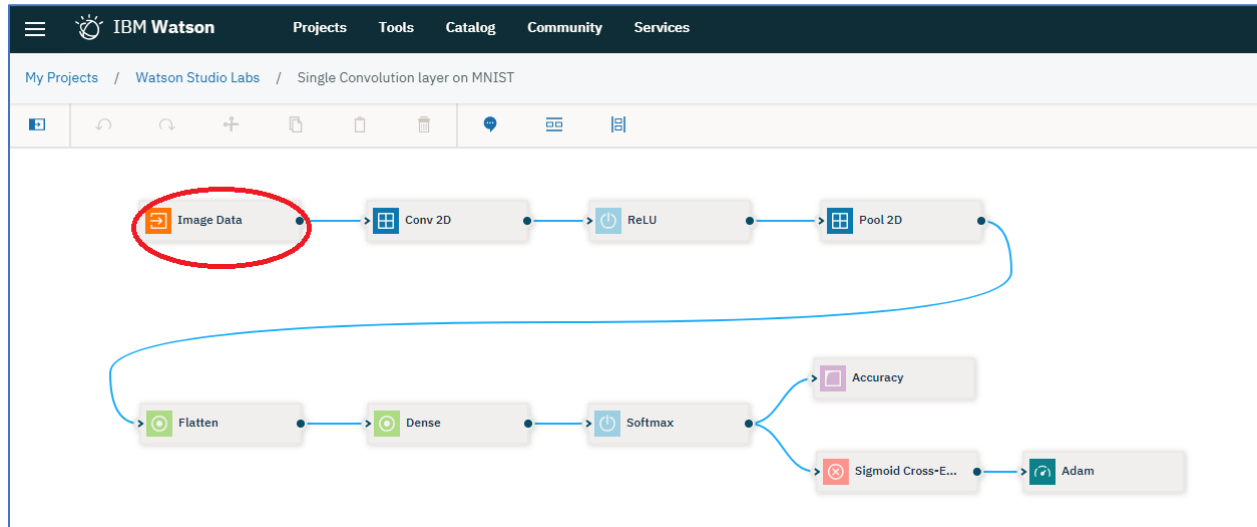
4. Click on the **Single Convolution Layer on MNIST** and then click on **Create**



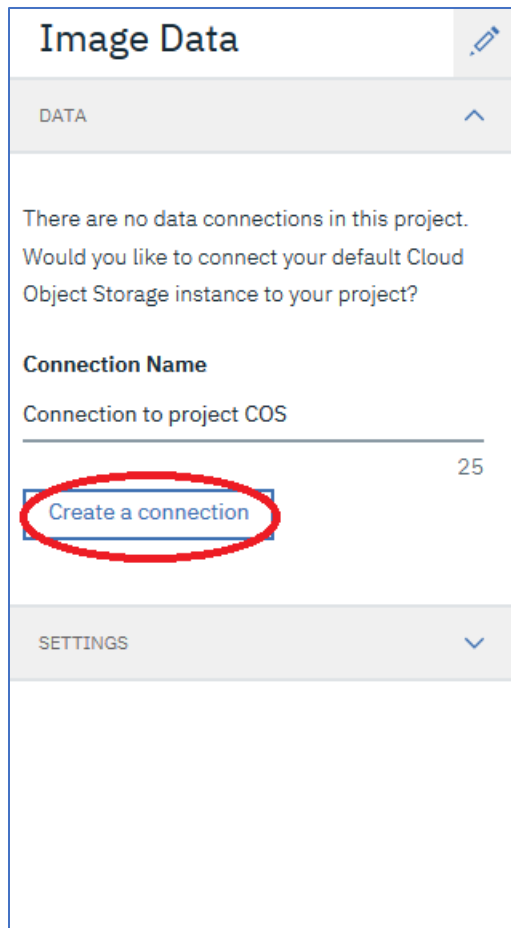
5. A standard convolution neural network (CNN) architecture is displayed on the Neural Network Modeler canvas. The Neural Network is represented in a graphical form instead of code. You will find a sidebar on the left of the screen containing all available neural

network components. The Neural Network Modeler enables the user to drag and drop nodes representing the different layers of a Neural Network and to connect them to create a **Flow**. Here, we have an already created neural network flow via the example.

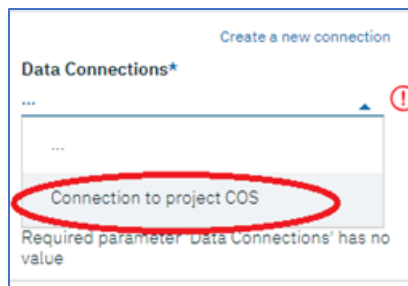
6. We still need to provide our neural network with a way to access the data we uploaded earlier. We configure the **Image Data** node for this purpose. Double-click on the **Image Data** node.



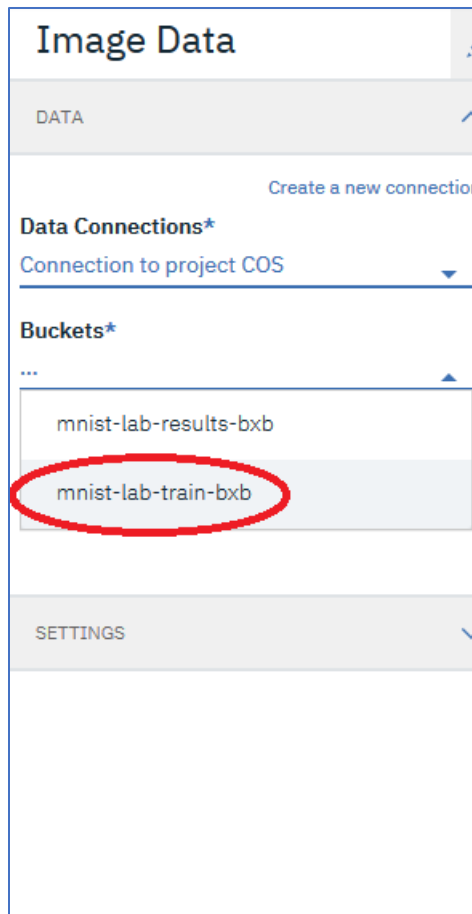
7. Leave the default **Connection Name** and click **Create Connection**.



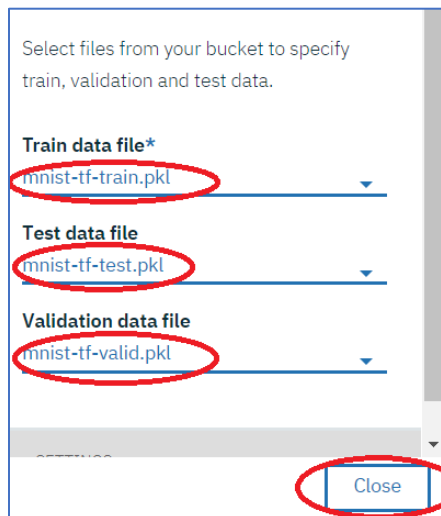
8. Click on the downward triangle icon ▼ underneath **Data Connections***. Click on the connection that was just created.





9. Click on the downward triangle icon ▼ underneath **Buckets***, and then click on the **mnist-lab-train-xxx** where “xxx” are your initials.



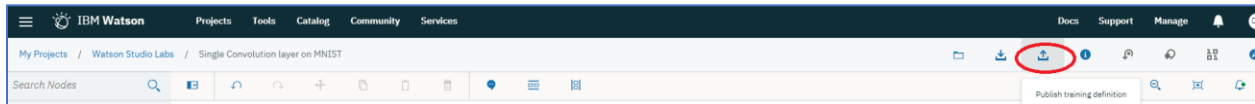
10. Click on the ▼ icon underneath **Train data file*** and select the **mnist-tf-train.pkl**. Assign the **Test data file** (**mnist-tf-test.pkl**), and **Validation data files** (**mnist-tf-valid.pkl**) in the same way and then click on **Close**.



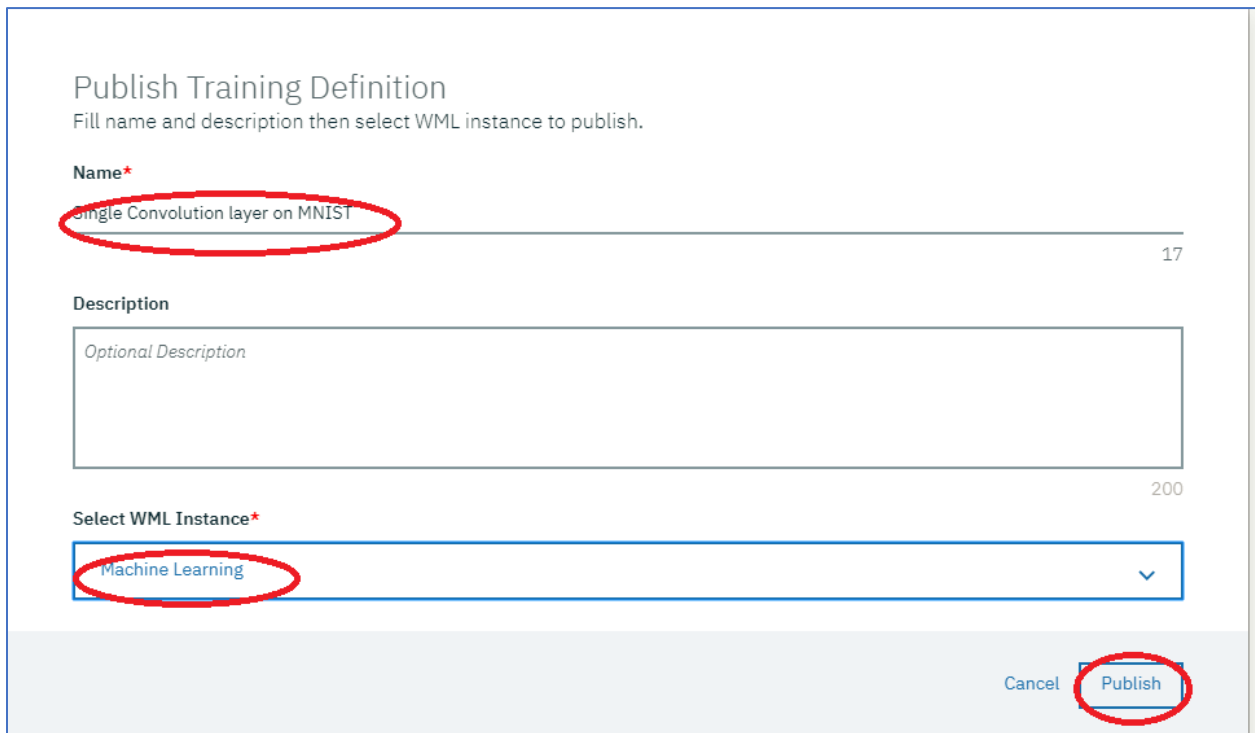
11. Explore the neural network flow modeler options

1. Click on the  icon to see the list of neural network component categories that are available
2. Explore the contents in each category. Hover over the components to get a pop-up description.
3. Click on the download icon  to see the multiple options for code generation.

12. Click on the **Publish** icon  to create a training definition.




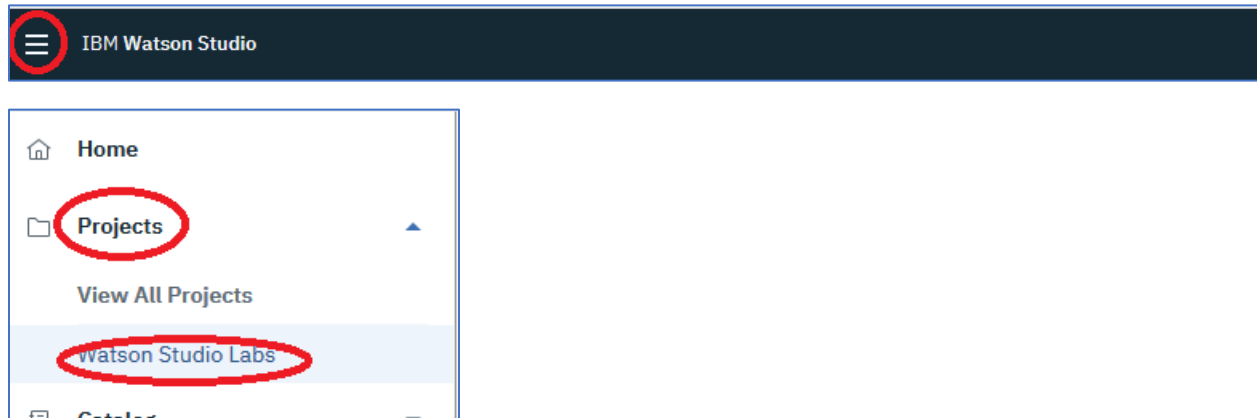
13. Enter a name for the training definition (or leave the default) and select the Machine Learning service that you created. Note, it will not be named Machine Learning unless that is the name that you used. Click on **Publish**.

A screenshot of the 'Publish Training Definition' form in IBM Watson Studio. The form has three main sections: 'Name*', 'Description', and 'Select WML Instance*'. The 'Name' field contains the text 'Single Convolution layer on MNIST' and is circled in red. The 'Description' field is empty and has a placeholder 'Optional Description'. The 'Select WML Instance' dropdown menu is open, showing 'Machine Learning' as the selected option, which is also circled in red. At the bottom right of the form, there are two buttons: 'Cancel' and 'Publish', with the 'Publish' button circled in red.

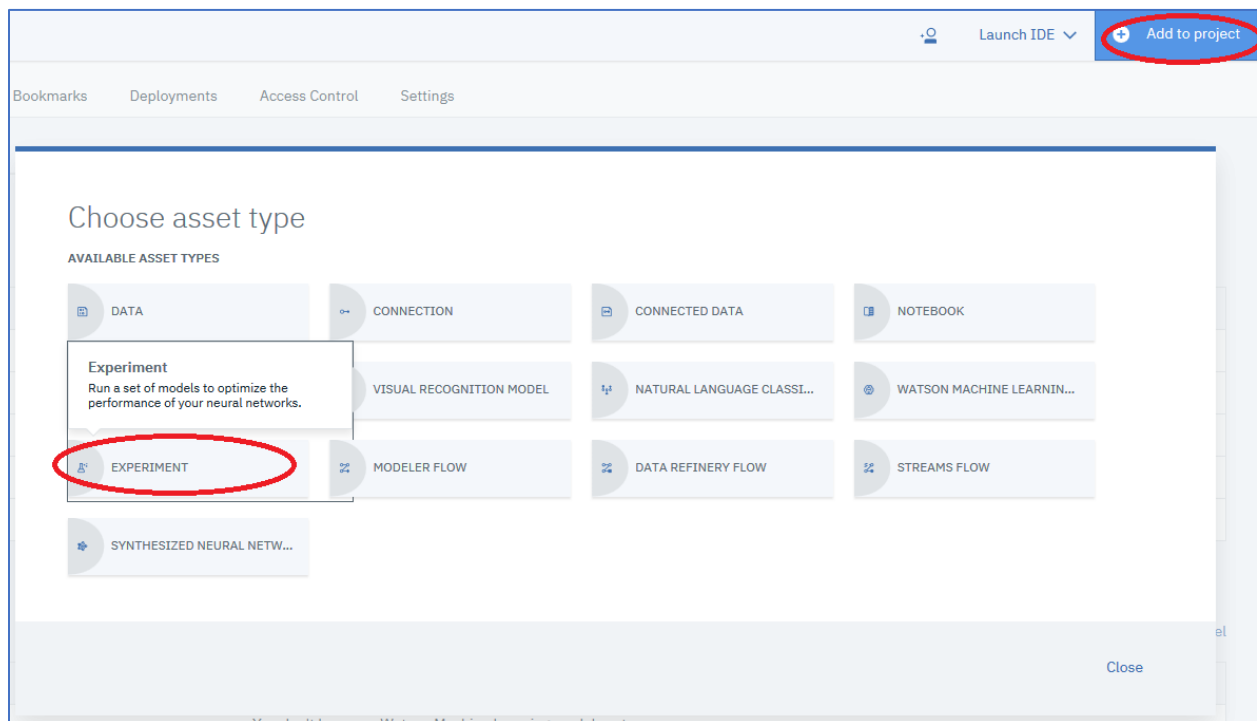
Step 3: Train the Model using Experiment Builder

As part of the model building process, we want to be able to compare different algorithms, and/or different algorithmic parameters to determine the best model. Experiment Builder is a facility in Watson Studio that supports this effort. Different training runs can be defined and run in parallel and their results can then be compared. In this lab, we will define only one training run to minimize the training time.

1. Return to the Watson Studio Labs **Assets** panel by clicking on the  icon, then click on **Projects**, and then **Watson Studio Labs**. Click on the **Assets** tab if the Assets panel is not displayed.



2. Click on **Add to project**, and then click **Experiment** to create a new Experiment.



3. Enter an Experiment **Name**, select the **Machine Learning** service, and then click on **Select** to assign a Cloud Storage bucket.

Define experiment details

Name
Single Convolutional Layer on MNIST

Description
Experiment description

Machine Learning Service
WatsonMachineLearning

Cloud Object Storage bucket for storing training data files
Select

Cloud Object Storage bucket for storing results
Select

Associate training definitions

Add training definition

Name	Compute plan
No training definitions associated.	

☐ Use global execution command (override training definition values)

4. Select **Existing connections**, and then select the **Connection to project COS** connection.

IBM Watson

Projects
Tools
Catalog
Community
Services

Cloud Object Storage bucket selection

Existing connections
New connection

Cloud Object Storage connection

Select Cloud Object Storage connection

Connection to project COS

5. We now need to assign the Training buckets. Select **Existing** underneath **Bucket containing training source data** and click on mnist-lab-train-xxx, where “xxx” are your initials. Click on **Select**.

Existing connections New connection

Cloud Object Storage connection

Connection to project COS

Bucket containing training source data

☒ Existing ☐ New

mnist-lab-train-bxb

Cancel **Select**

6. Click on **Select** underneath **Cloud Object Storage** for storing results.

New experiment BETA

Define experiment details

Name

Single Convolution Layer on MNIST

67

Description


Experiment description

300

Machine Learning Service

Machine Learning

Cloud Object Storage bucket for storing training source files

Source: Connection to project COS / mnist-lab-train-bxb  [Update](#)

Cloud Object Storage bucket for storing results

Select

7. Follow the same procedure used to assign the training bucket to assign the results bucket. Assign bucket mnist-lab-results-xxx, where “xxx” are your initials, and then click on **Select**.

Cloud Object Storage results bucket selection

Existing connections | New connection

Cloud Object Storage connection
Connection to project COS

Bucket containing results data
☒ Existing ☐ New

mnist-lab-results-bno

Cancel Select

8. We now need to associate a Training Definition. Click on **Add Training Definition**.

IBM Watson | Projects | Tools | Catalog | Community | Services | Docs | Support | Manage

New experiment BETA

Define experiment details

Name
Single Convolution Layer on MNIST

Description
Experiment description

Machine Learning Service
Machine Learning

Cloud Object Storage buckets for storing training source and results files
Source: Connection to project COS / mnist-lab-3-train...
Results: Connection to project COS / mnist-lab-3-resul...
Update

Associate training definitions

Add training definition

NAME	COMPUTE PLAN
No training definitions associated.	

☐ Use global execution command (override training definition values)

9. Click on **Existing training definition**, and select **Single Convolution Layer on MNIST**, select **1/2 x NVIDIA Tesla K80 (1 GPU)** for the compute plan, and then click **Select**.

Add training definition

New training definition

Select training definition

Existing training definitions

Single Convolution layer on MNIST

Training definition attributes

Compute plan

1/2 x NVIDIA® Tesla® K80 (1 GPU)

Hyperparameter optimization method

None

Cancel **Create**

10. Click **Create and run**.

New experiment BETA

Define experiment details

Name

Single Convolution Layer on MNIST

67

Description

Experiment description

300

Machine Learning Service

Machine Learning

Cloud Object Storage buckets for storing training source and results files

Source: Connection to project COS / mnist-lab-3-train... [🔗](#)

Results: Connection to project COS / mnist-lab-3-resul... [🔗](#) [Update](#)

If your connection is authorized for dashboard access, click the bucket name above to launch the dashboard. It may take a few seconds for the dashboard link to work for newly created buckets. Alternatively, reference the Cloud Object Storage APIs [🔗](#).

Associate training definitions

[+ Add training definition](#)

NAME	COMPUTE PLAN
Single Convolution layer on MNIST	1/2 x NVIDIA® Tesla® K80 (1 GPU)

☐ Use global execution command (override training definition values)

Cancel **Create and run**

Step 4: Monitor the Training Progress and Results

Training runs will be first queued, then in-process, and then completed. Use the **Training Runs** tab to keep track of progress. Usually, it will take between 3-5 minutes.

Queued Status

My Projects / Watson Studio Labs / Single Convolution Layer on MNIST

Single Convolution Layer on MNIST

Cancel runs in progress Add training runs

Training Runs Compare Runs Overview

1
Runs in total

0 hr, 0 min, 0 sec
Total running time

Queued

NAME	SUBMITTED
Single Convolution layer on MNIST	0 hr, 0 min, 6 sec ago

In progress

NAME	DURATION
No training runs found.	

Completed

NAME	STATUS	DURATION	ACTIONS
No training runs found.			

In-Process Status

My Projects / Watson Studio Labs / Single Convolution Layer on MNIST

Single Convolution Layer on MNIST

Cancel runs in progress Add training runs

Training Runs Compare Runs Overview

1
Runs in total

0 hr, 0 min, 43 sec
Total running time

Queued

NAME	SUBMITTED
No training runs found.	

In progress

NAME	DURATION
Single Convolution layer on MNIST	0 hr, 0 min, 43 sec

Completed

NAME	STATUS	DURATION	ACTIONS
No training runs found.			

Completed Status – Note the statistics on the accuracy of the training set and validation set.

Single Convolution on MNIST

Add training runs

Training Runs Compare Runs Overview

1
Runs in total

0 hr, 2 min, 13 sec
Total running time

Queued

NAME	SUBMITTED
No training runs found.	

In progress

NAME	DURATION
No training runs found.	

Completed

NAME	STATUS	DURATION	ACC	LOSS	VAL_ACC	VAL_LOSS	ACTIONS
Single Convolution layer on MNIST	completed	0 hr, 2 min, 13 sec	0.994	0.019	0.976	0.108	

1. When completed, click on the Single Convolution layer on MNIST link.

My Projects / Watson Studio Labs / Single Convolution Layer on MNIST

Single Convolution Layer on MNIST

Training Runs Compare Runs Overview

1 0 hr, 2 min, 45 sec
Runs in total Total running time

Queued

NAME	SUBMITTED
No training runs found.	

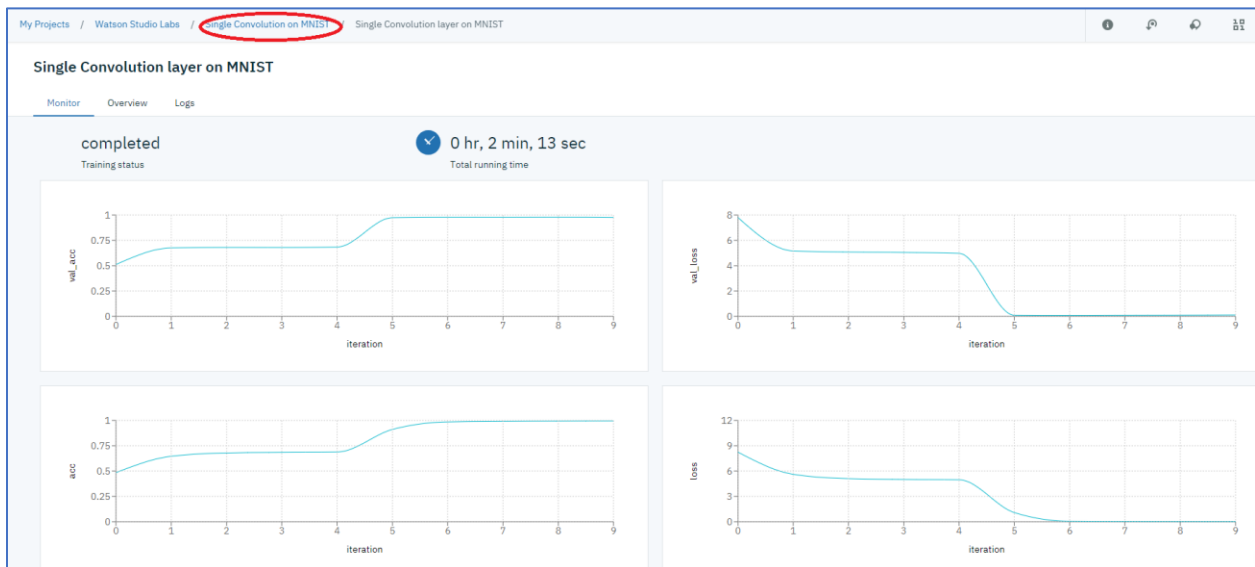
In progress

NAME	DURATION
No training runs found.	

Completed

NAME	STATUS	DURATION	ACTIONS
Single Convolution layer on MNIST	completed	0 hr, 2 min, 45 sec	

2. The display of the statistics over the training iterations is displayed. Click on the Single Convolution on MNIST tab to return to the Training Runs screen.



Step 5: Save and Deploy the Trained Model

We will now save the trained model to the Watson Machine Learning repository.

1. Click on the vertical ellipse under ACTIONS and click **Save model**.

Completed

NAME	STATUS	DURATION	ACTIONS
Single Convolution layer on MNIST	completed	0 hr, 2 min, 45 sec	

2. Enter a **Name** for the model (Single Convolution layer on MNIST) and click **Save**.

Save Model

Name
Single Convolutional layer on MNIST 65

Description
Model description 300

Cancel Save

3. Return to the Watson Studio **Assets** panel, by clicking on **Watson Studio Labs** in the breadcrumb path. Click on the **Assets** tab if the Assets panel is not showing.

IBM Watson Projects Tools Catalog Community Services

My Projects / Watson Studio Labs Single Convolution Layer on MNIST

✓ Model successfully saved. View model details [here](#).

4. Click on the newly saved model

Models New model

NAME	STATUS	TYPE	RUNTIME	LAST MODIFIED	ACTIONS
Single Convolutional Layer on MNIST	trained	tensorflow-1.5	python-3.5	5 Jun 2018	

5. Click on **Deployments**.

The screenshot shows the IBM Watson Studio interface. The top navigation bar includes 'My Projects', 'Watson Studio Labs', and 'Single Convolutional layer on MNIST...'. The main header shows 'Single Convolutional layer on MNIST' with a sub-header containing 'Overview', 'Evaluation', and 'Deployments' (circled in red). Below this is a 'Summary' section with a table of model details.

Property	Value
Machine learning service	Machine Learning
Model Type	tensorflow-1.5
Runtime environment	python-3.5
Training date	5 Jun 2018, 3:48 PM
Latest version	1c472928-e0ac-4146-9985-5b1c02bb8881

6. Click on **Add Deployment**.

This screenshot shows the 'Deployments' tab of the IBM Watson Studio interface. The 'Add Deployment' button, located in the top right corner of the deployment list area, is circled in red. Below the button is a table with columns for NAME, STATUS, DEPLOYMENT TYPE, and ACTIONS. The table currently shows a message: 'Your model is not deployed.'

7. Enter a **Name** (e.g. Single Convolution layer on MNIST Deployed), select **Web Service** (should be the default), and click on **Save**.

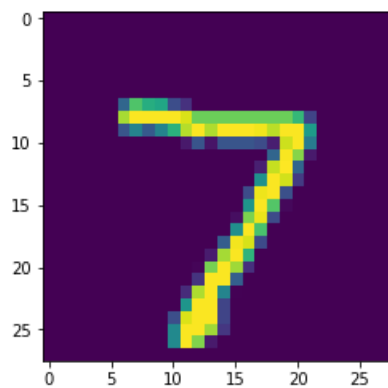
The screenshot shows the 'Create Deployment' form. The 'Name' field is filled with 'Single Convolutional layer on MNIST Deployed' and is circled in red. The 'Description' field is empty. Under the 'Deployment type' section, 'Web service' is selected with a radio button and is circled in red. At the bottom right, the 'Save' button is circled in red.

8. The model is successfully deployed.

Overview Evaluation Deployments Lineage			
			+ Add Deployment
NAME	STATUS	DEPLOYMENT TYPE	ACTIONS
Single Convolution Layer on MNIST Deployed	DEPLOY_SUCCESS	Web Service	⋮

Step 6: Test the Deployed Model

We will now test the deployed model using the sample image data contained in the file test.json that was extracted from the mnist.zip file previously. The test.json file is a representation of the following image.



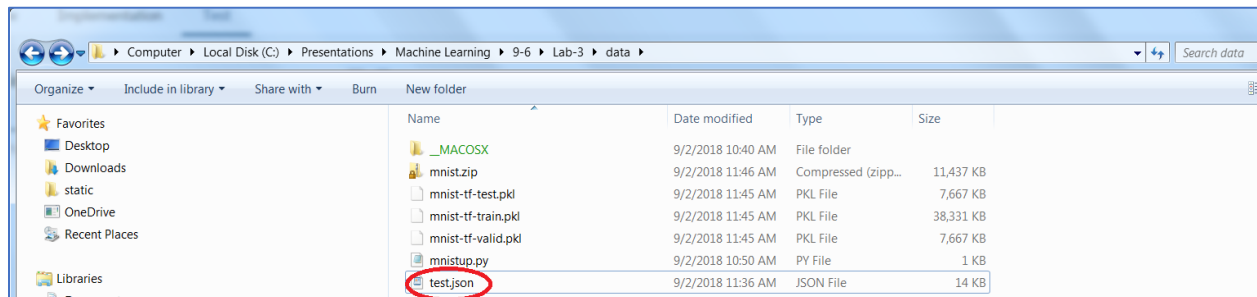
1. Click on the vertical ellipse, and then click on **View**.

Single Convolution on MNIST			
Overview Evaluation Deployments Lineage			
			+ Add Deployment
NAME	STATUS	DEPLOYMENT TYPE	ACTIONS
Single Convolution Layer on MNIST Deployed	DEPLOY_SUCCESS	Web Service	⋮
			View
			Delete

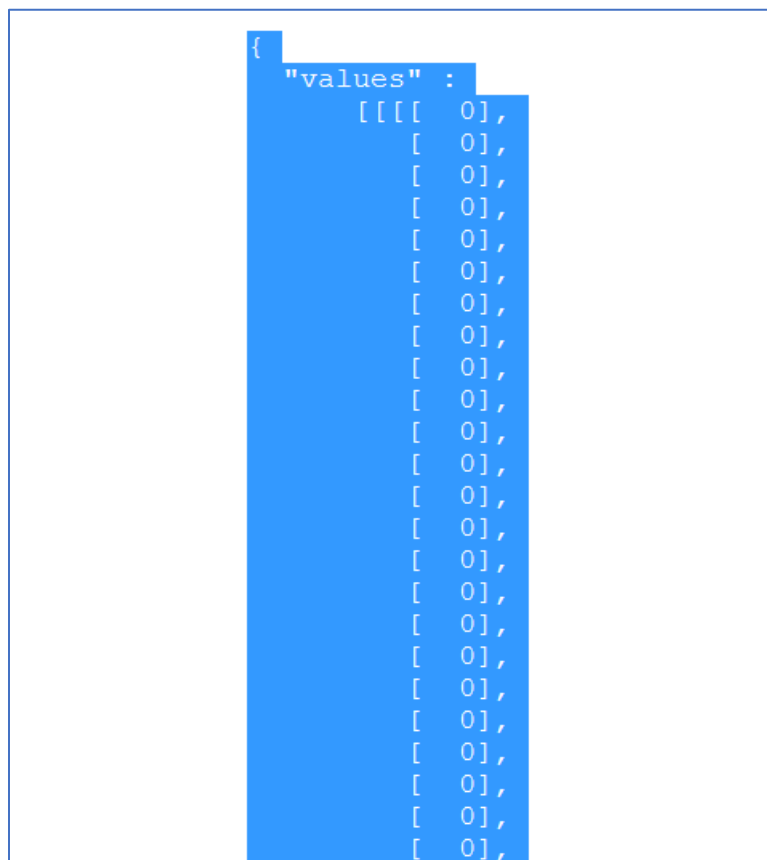
2. Click on **Test**.

Single Convolution Layer on MNIST Deployed	
Overview Implementation Test	
Deployment	
Name	Single Convolution Layer on MNIST Deployed
Type	Web Service
Deployment ID	89cf10e5-bd95-4b56-a728-e8b65fa98d83

- Go to the file directory where you have the “test.json” file stored, and double-click on the file.





4. Select the contents of the file by placing the cursor to the left of the { and pressing and holding the <Shift><Ctrl><End> keys.



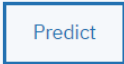
- Copy and paste the content into the **Paste the request payload here** input data section. Make sure you have both the top bracket { at the beginning of the input data section and the bottom bracket } at the end of the input section data section, and then click on **Predict**.

Single Convolution Layer on MNIST Deployed

Overview Implementation **Test**



Enter input data  

```
{  
  "values":  
    [[[[ 0],  
        [ 0],  
        [ 0],  
        [ 0],  
        [ 0],  
        [ 0],  
        [ 0],  
        [ 0]]]]  
}
```

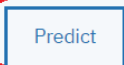


Single Convolution Layer on MNIST Deployed

Overview Implementation **Test**

Enter input data  



```
{  
  "values":  
    [[[[ 0],  
        [ 0],  
        [ 0],  
        [ 0],  
        [ 0],  
        [ 0],  
        [ 0],  
        [ 0]]]]  
}
```



6. Each of the values represent the confidence level for the digits 0,1,2,3,4,5,6,7,8, and 9. The digit that is recognized would have the largest of the confidence levels. Based on the values returned, we can see that the number 7 would be selected as the best fit for this sample image which matches the image shown above.

Single Convolution Layer on MNIST Deployed

[Overview](#)[Implementation](#)[Test](#)

Enter input data

```
{  
  "values":  
    [[[[ 0],  
        [ 0],  
        [ 0],  
        [ 0],  
        [ 0],  
        [ 0],  
        [ 0]]]]  
}
```

Predict

```
{  
  "fields": [  
    "prediction"  
  ],  
  "values": [  
    [  
      1.3815683352121771e-15,  
      2.1411425399327925e-18,  
      1.4536198037363307e-13,  
      1.547869092014681e-13,  
      3.484191859962678e-18,  
      1.3316728982702908e-19,  
      3.832952076167682e-23,  
      1,  
      1.7163898882906203e-13,  
      4.053319832553193e-11  
    ]  
  ]  
}
```

You have completed the Lab!

- ✓ Uploaded the data files to IBM Cloud Storage.
- ✓ Designed the neural network
- ✓ Trained the model
- ✓ Monitored the training progress and results
- ✓ Saved and Deployed the Trained Model
- ✓ Tested the Deployment