

Lab Center – Hands-on Lab

Session #3259

Session Title: Hands-on Introduction to Data Science using Watson Studio

Lab-3 - Build, Train, Save, Deploy and Test a Convolutional Neural Network Model using MNIST

Bernard Beekman, IBM , beekmanb@us.ibm.com

Michael Cronk, IBM, michael.cronk@ibm.com

Table of Contents

Disclaimer	3
Introduction	5
Step 1: Set up the Data Files in IBM Cloud Storage	5
Step 2: Design the Neural Network and Publish Training Definition	8
Step 3: Train the Model using Experiment Builder	13
Step 4: Monitor the Training Progress and Results	18
Step 5: Save and Deploy the Trained Model.....	20
We Value Your Feedback!	27

Disclaimer

IBM's statements regarding its plans, directions, and intent are subject to change or withdrawal without notice at IBM's sole discretion. Information regarding potential future products is intended to outline our general product direction and it should not be relied on in making a purchasing decision.

The information mentioned regarding potential future products is not a commitment, promise, or legal obligation to deliver any material, code or functionality. Information about potential future products may not be incorporated into any contract.

The development, release, and timing of any future features or functionality described for our products remains at our sole discretion I/O configuration, the storage configuration, and the workload processed. Therefore, no assurance can be given that an individual user will achieve results like those stated here.

Information in these presentations (including information relating to products that have not yet been announced by IBM) has been reviewed for accuracy as of the date of initial publication and could include unintentional technical or typographical errors. IBM shall have no responsibility to update this information. **This document is distributed "as is" without any warranty, either express or implied. In no event, shall IBM be liable for any damage arising from the use of this information, including but not limited to, loss of data, business interruption, loss of profit or loss of opportunity.** IBM products and services are warranted per the terms and conditions of the agreements under which they are provided.

IBM products are manufactured from new parts or new and used parts.

In some cases, a product may not be new and may have been previously installed. Regardless, our warranty terms apply."

Any statements regarding IBM's future direction, intent or product plans are subject to change or withdrawal without notice.

Performance data contained herein was generally obtained in controlled, isolated environments. Customer examples are presented as illustrations of how those customers have used IBM products and the results they may have achieved. Actual performance, cost, savings or other results in other operating environments may vary.

References in this document to IBM products, programs, or services does not imply that IBM intends to make such products, programs or services available in all countries in which IBM operates or does business.

Workshops, sessions and associated materials may have been prepared by independent session speakers, and do not necessarily reflect the views of IBM. All materials and discussions are provided for informational purposes only, and are neither intended to, nor shall constitute legal or other guidance or advice to any individual participant or their specific situation.

It is the customer's responsibility to insure its own compliance with legal requirements and to obtain advice of competent legal counsel as to the identification and interpretation of any relevant laws and regulatory requirements that may affect the customer's business and any actions the customer may

need to take to comply with such laws. IBM does not provide legal advice or represent or warrant that its services or products will ensure that the customer follows any law.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products about this publication and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products. IBM does not warrant the quality of any third-party products, or the ability of any such third-party products to interoperate with IBM's products. **IBM expressly disclaims all warranties, expressed or implied, including but not limited to, the implied warranties of merchantability and fitness for a purpose.**

The provision of the information contained herein is not intended to, and does not, grant any right or license under any IBM patents, copyrights, trademarks or other intellectual property right.

IBM, the IBM logo, ibm.com and [names of other referenced IBM products and services used in the presentation] are trademarks of International Business Machines Corporation, registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the Web at "Copyright and trademark information" at: www.ibm.com/legal/copytrade.shtml.

© 2019 International Business Machines Corporation. No part of this document may be reproduced or transmitted in any form without written permission from IBM.

U.S. Government Users Restricted Rights — use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM.

Introduction

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

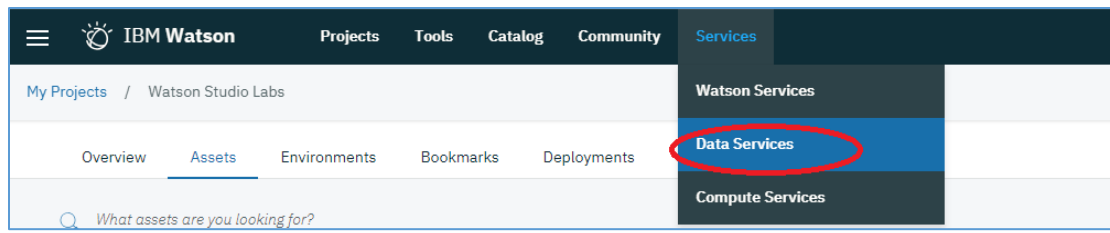
Step 1: Set up the Data Files in IBM Cloud Storage

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 3 files - a training file (mnist-tf-train.pkl), test file (mnist-tf-test.pkl), and a validation file (mnist-tf-valid.pkl) in pickle format.



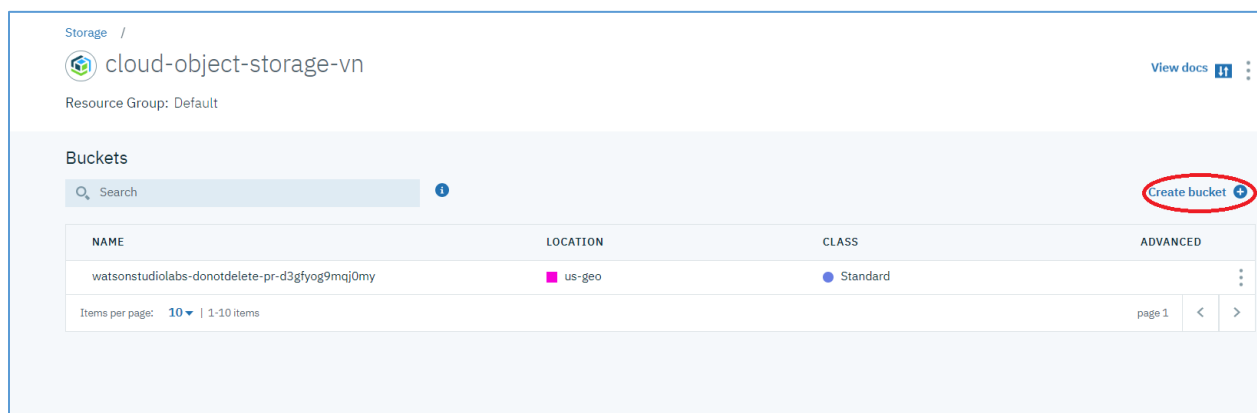
2. Select the **Services** tab and the click on **Data Services**



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. Click on **Create Buckets**



5. Enter a unique name for the bucket - mnist-lab-3-train-xxx (replace xxx with your initials), click on **Cross-Region** for the Resiliency and click **Create**.

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

mnist-lab-3-train-bxb

Resiliency: **Cross Region** Location: **us-geo**

Storage class: **Standard**

☐ Add Key Protect Keys (optional)

The feature is currently not supported in the location you have selected. [Learn more](#)

☐ Add archive policy (optional)

The feature is currently not supported in the location you have selected. [Learn more](#)

Create bucket

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

Objects

Search

Object Name Size Last Modified

Drag and drop files or folders to upload them.

Computer > Local Disk (C:) > Presentations > Machine Learning > 9-6 > Lab-3 > data

Name	Date modified	Type	Size
mnist.zip	6/5/2018 10:38 AM	Compressed (zipp...	11,350 KB
mnist-tf-test.pkl	9/1/2018 11:59 AM	PKL File	7,667 KB
mnist-tf-train.pkl	9/1/2018 11:59 AM	PKL File	38,331 KB
mnist-tf-valid.pkl	9/1/2018 11:59 AM	PKL File	7,667 KB

- Click on Buckets to add a second bucket.

Getting started

Buckets

Storage / cloud-object-storage-vn / mnist-lab3-blb-train-1

mnist-lab3-blb-train-1

View docs

Objects

Search

Upload

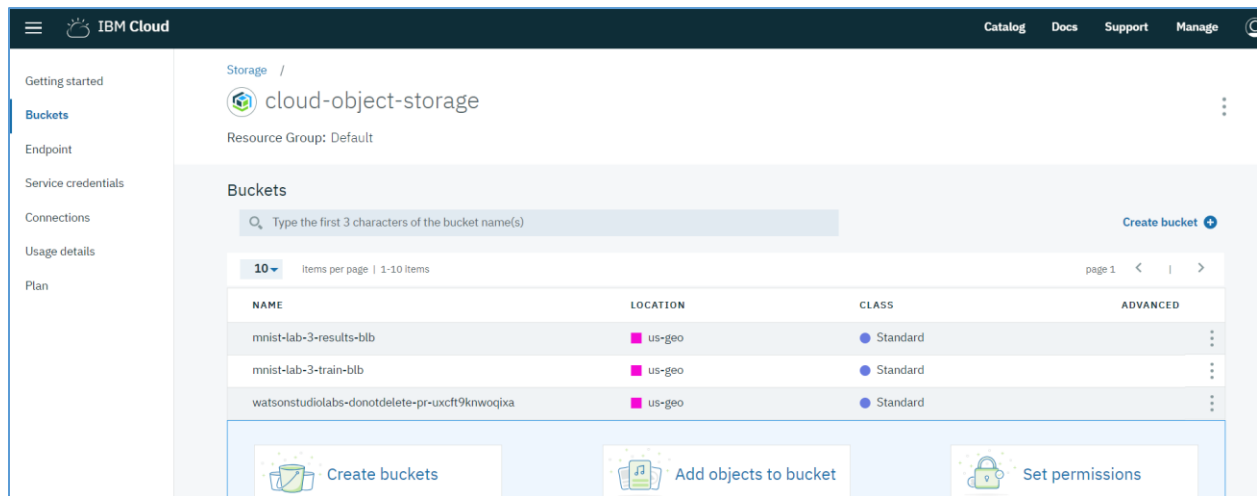
Object Name	Size	Last Modified
mnist-tf-test.pkl	7.5 MB	09/01/2018 1:38:13 PM
mnist-tf-train.pkl	37.4 MB	09/01/2018 1:39:38 PM
mnist-tf-valid.pkl	7.5 MB	09/01/2018 1:39:55 PM

Items per page: 10 | 1-10 items

page 1 < >

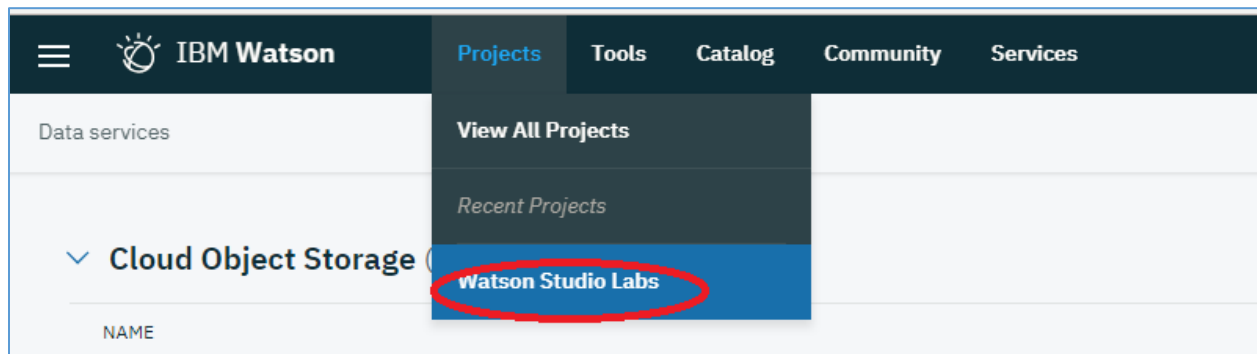
- Name it mnist-lab-3-results-xxx, where xxx are your initials. Follow the procedure above to create the second bucket. No files need to be added.

9. The Cloud Object Storage panel should appear as below.

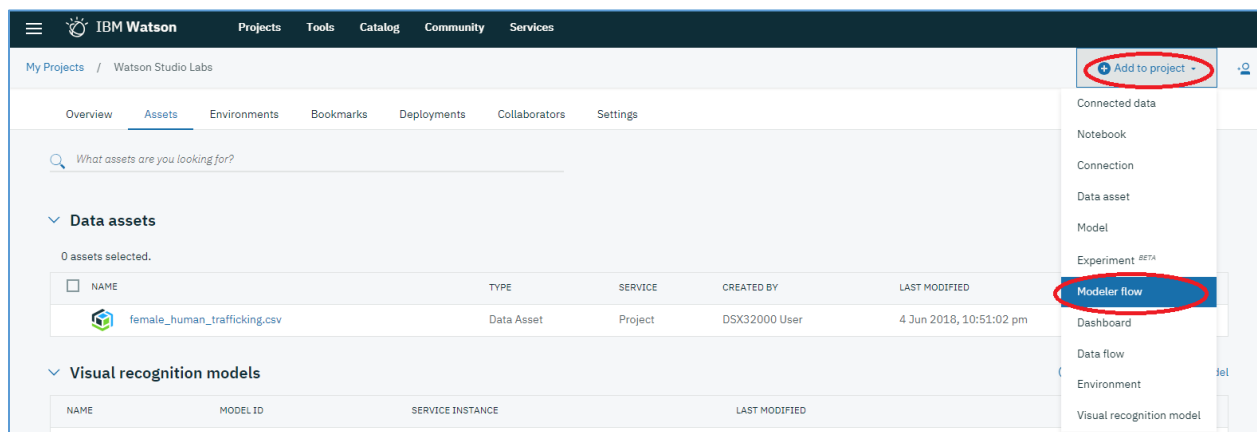


Step 2: Design the Neural Network and Publish Training Definition

1. Return to Watson Studio, and click on the **Projects** tab, and **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**

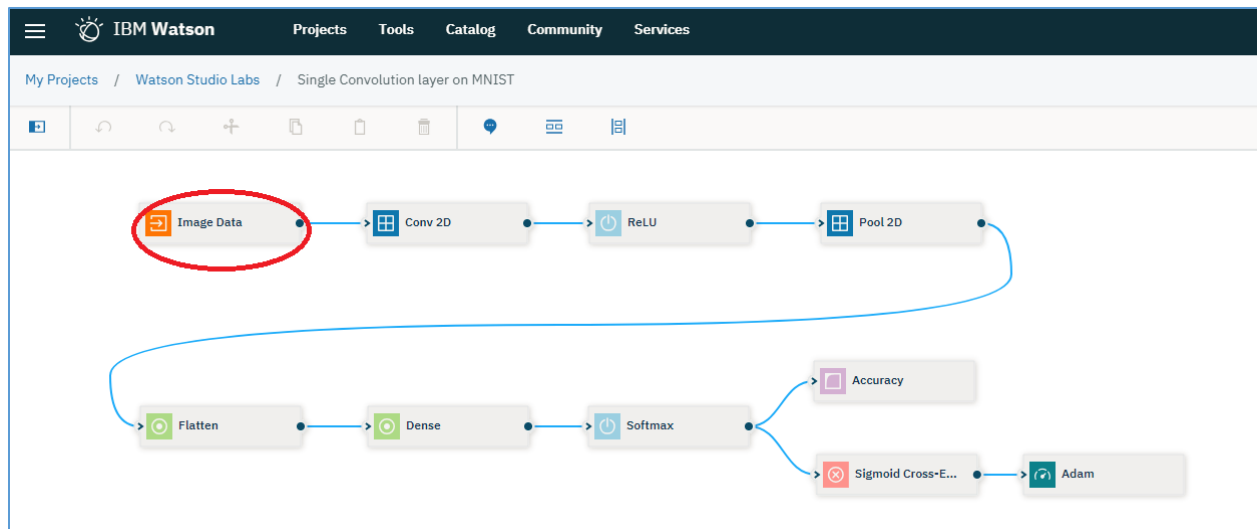
New From file From example

Select one of the samples below to get started with an existing stream that suits the kind of modeling you want to do. When you create the stream it will be added to your project, allowing you to modify and save your changes.

<p>SPSS MODELER Drug Study Example Use neural network and C5.0 algorithms to build classification models that allow you to predict the correct type of drug for a patient based on various health metrics.</p>	<p>SPSS MODELER Sales Promotion Study Use neural network and C5.0 algorithms to predict the effect of advertising promotions on the sale of various items. Input data of sales before and after a sales promotion are used to train the model to predict the effectiveness of advertising.</p>	<p>NEURAL NETWORK DESIGN Single Convolution layer on MNIST Classification of MNIST dataset with a simple deep Convolutional Neural Network containing one single convolutional and dense layer.</p>	<p>NEURAL NETWORK DESIGN CNN on CIFAR10 Classification of CIFAR-10 dataset with a simple deep Convolutional Neural Network.</p>
--	--	---	---

Cancel **Create**

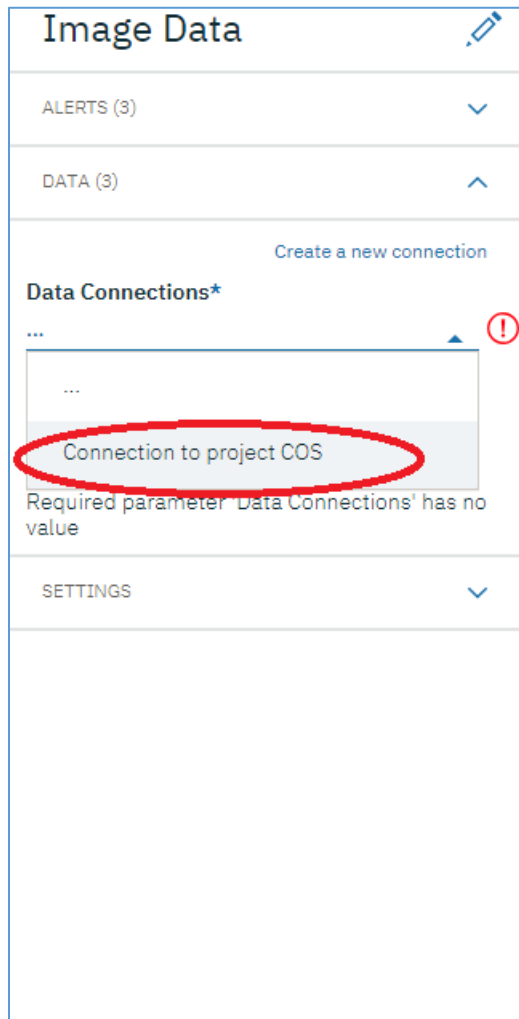
5. A standard convolution neural network (CNN) architecture is displayed. We need to configure the Image Data node. Double-click on Image Data.



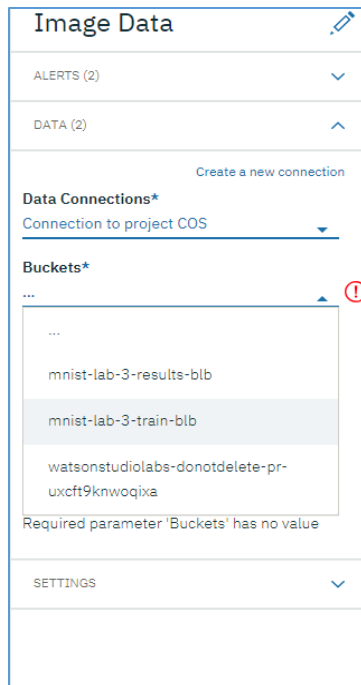
6. Optionally change the default **Connection Name**, and then click on **Create a connection**.

The screenshot shows the 'Image Data' configuration panel. The 'DATA (2)' section is expanded, showing a message: 'There are no data connections in this project. Would you like to connect your default COS instance to your project?'. Below this, the 'Connection Name' field is set to 'Connection to project COS'. The 'Create a connection' button is circled in red.

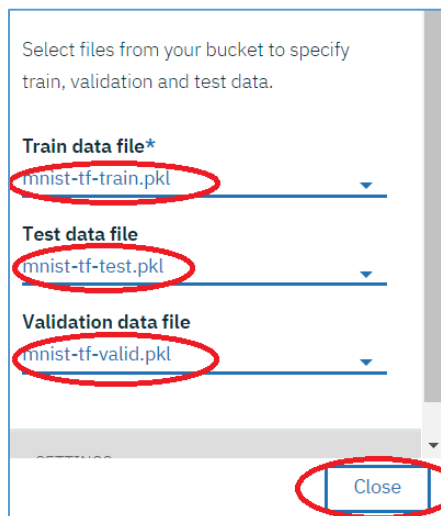
7. Click on the downward triangle icon ▼ underneath **Data Connections***. Click on the connection that was just created.





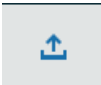
8. Click on the downward triangle icon ▼ underneath **Buckets***, and then click on the **mnist-lab-3-train-xxx** where “xxx” are your initials.

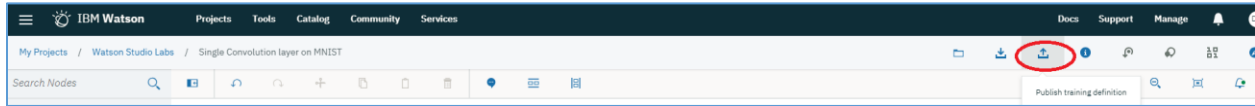


9. Click on the ▼ icon underneath **Train data file*** and select the **mnist-tf-train-pkl**. Assign the Test data file(mnist-tf-test-pkl), and Validation data files(mnist-tf-valid-pkl) in the same way and then click on **Close**.

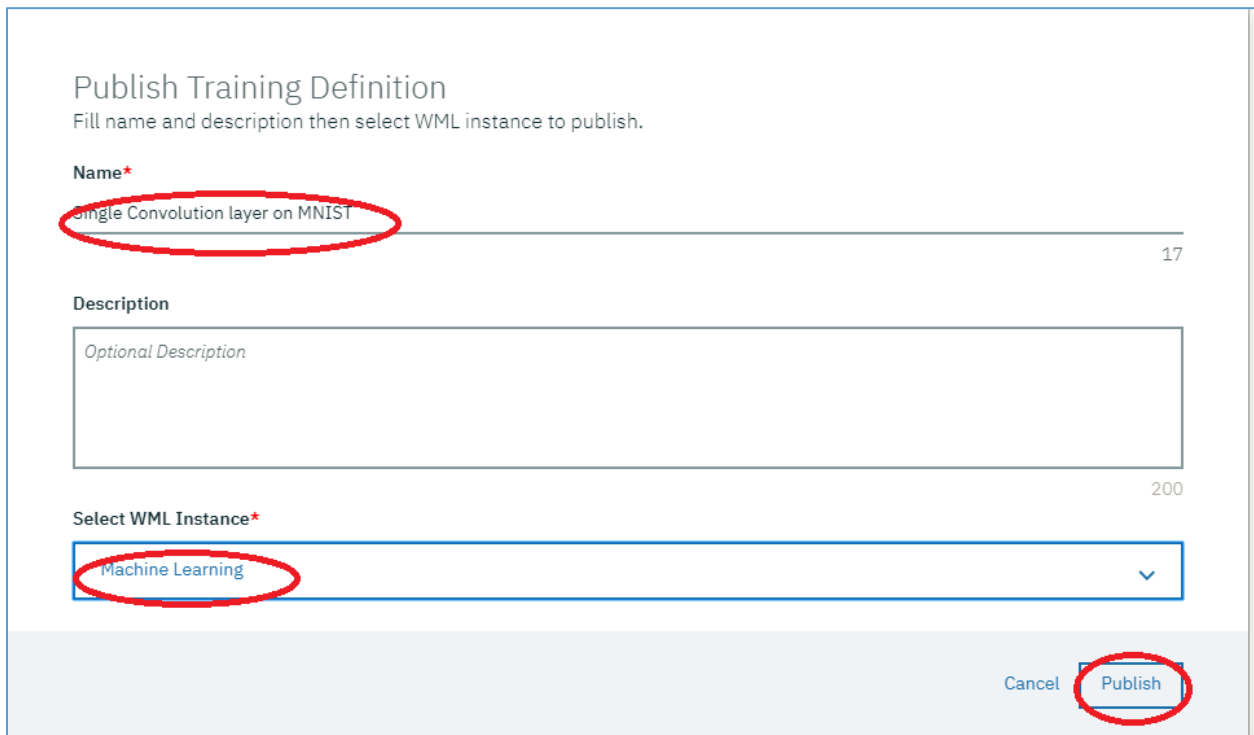


10. Explore the neural network flow modeler options
 1. Click on the  icon to see the list of neural network component categories that are available
 2. Explore the contents in each category. Hover over the components to get a pop-up description.
 3. Drag some nodes on the canvas and double-click to see the parameters. **Note remove these nodes before doing step 11.**

4. Click on the download icon  to see the multiple options for code generation.
11. Click on the **Publish** icon  to create a training definition.



12. 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.

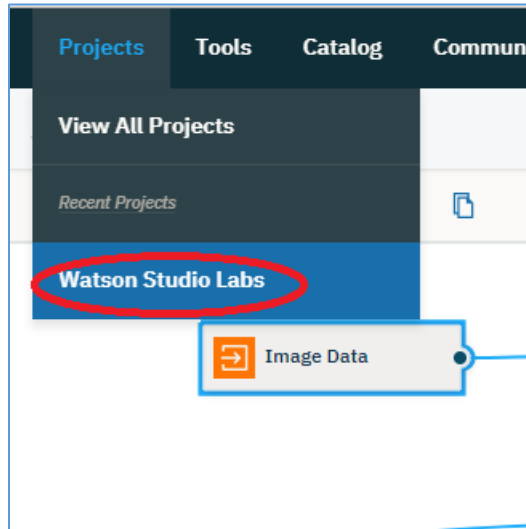
A screenshot of the 'Publish Training Definition' form in IBM Watson Studio. The form has three main sections: 'Name*', 'Description', and 'Select WML Instance*'. The 'Name' field contains 'Single Convolution layer on MNIST' and is circled in red. The 'Description' field is empty and has a placeholder '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, there are 'Cancel' and 'Publish' buttons, with the 'Publish' button circled in red. The form also shows character counts of 17 for the name and 200 for the description.

Step 3: Train the Model using Experiment Builder

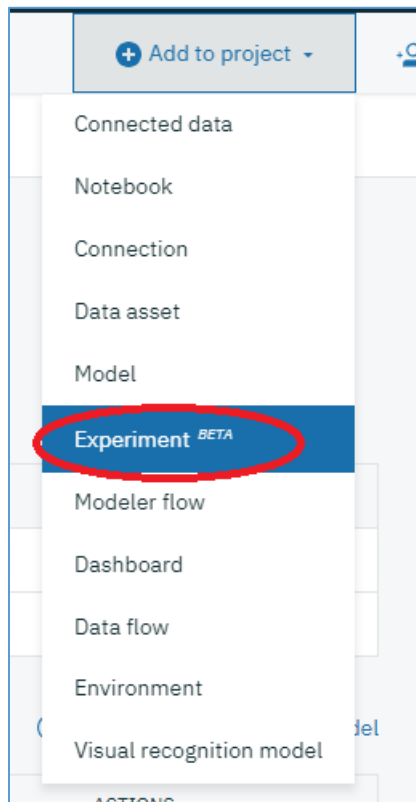
As part of the model building process, we want to be able to compare different algorithms, and/or different algorithmic parameters to determine the best model. Experiment Builder is a facility in Watson Studio that supports this effort. Different training runs can be defined and run

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 **Projects** tab 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.

New experiment ^{BETA}

Define experiment details

Name
Single Convolution Layer on MNIST

Description
Experiment description

Machine Learning Service
Machine Learning

Cloud Object Storage bucket for storing training source and results files
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 and Results buckets. Select **Existing** underneath **Bucket containing training data**, and click on mnist-lab-3-train-xxx, where “xxx” are your initials. Click on **Select**.

Cloud Object Storage training source bucket selection

Existing connections New connection

Cloud Object Storage connection

Connection to project COS

Bucket containing training source data

Existing ☒ New ☐

mnist-lab-3-train-bdb-3

Cancel Select

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

New experiment ^{BETA}

Define experiment details

Name

Single Convolution on MNIST

73

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-3-train-b... [Link](#) 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-3-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-3-results-bxb-1

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 **existing 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... [\[icon\]](#)

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

Associate training definitions

[+ Add training definition](#)

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

☐ 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.

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.	

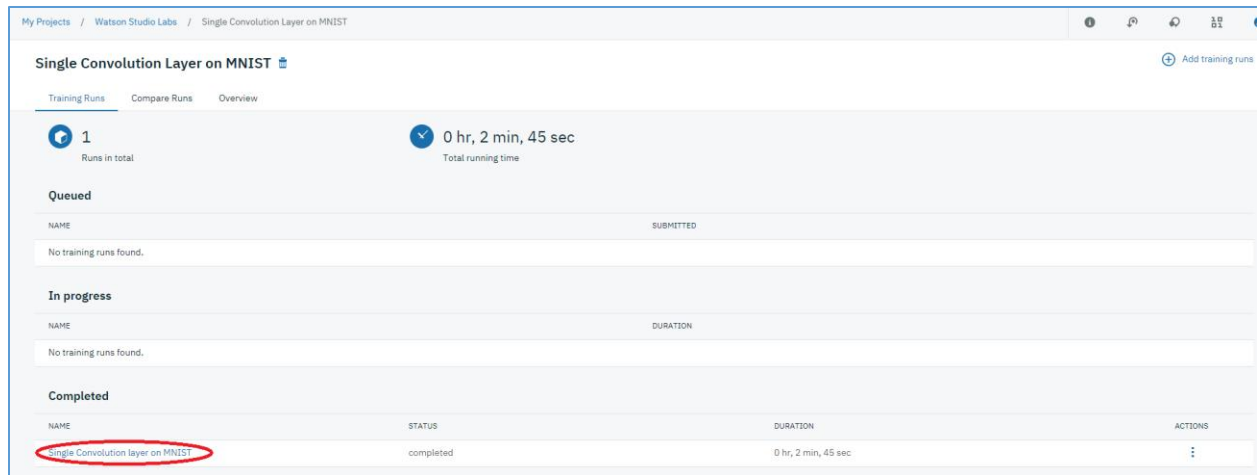
In progress

NAME	DURATION
No training runs found.	

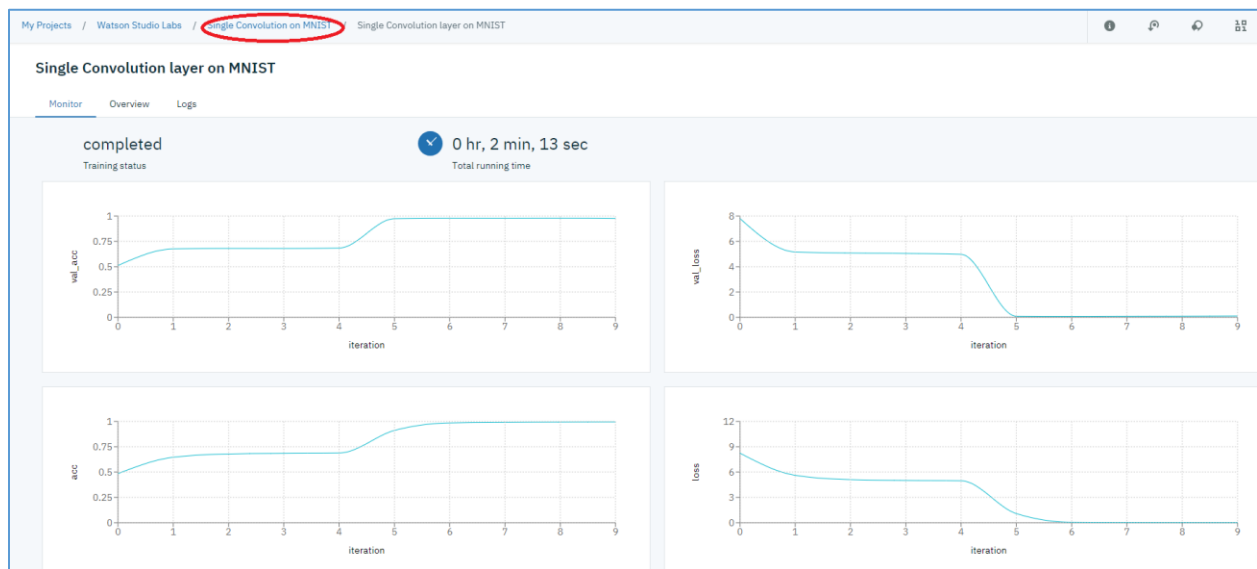
Completed

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. Click on the Single Convolution layer on MNIST link.



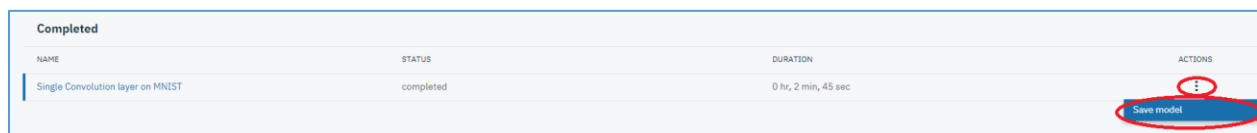
2. The display of the statistics over the training iterations is displayed. Click on the Single Convolution on MNIST tab to return to the Training Runs screen.



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**.



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

- 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](#).

- Click on the newly saved model

Models New model

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

- 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:

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	1c472926-e0ac-4146-9985-5b1c02bb8881

6. Click on **Add Deployment**.

This screenshot shows the 'Deployments' tab of the same project. The 'Add Deployment' button in the top right corner is circled in red. Below the button is a table with columns: NAME, STATUS, DEPLOYMENT TYPE, and ACTIONS. The table currently contains one row with the text '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 'Deployment type', the 'Web service' radio button is selected and 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.

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



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

Computer > Local Disk (C:) > Presentations > Machine Learning > 9-6 > Lab-3 > data				
Organize	Include in library	Share with	Burn	New folder
Name	Date modified	Type	Size	
._MACOSX	9/2/2018 10:40 AM	File folder		
mnist.zip	9/2/2018 11:46 AM	Compressed (zipp...	11,437 KB	
mnist-tf-test.pkl	9/2/2018 11:45 AM	PKL File	7,667 KB	
mnist-tf-train.pkl	9/2/2018 11:45 AM	PKL File	38,331 KB	
mnist-tf-valid.pkl	9/2/2018 11:45 AM	PKL File	7,667 KB	
mnistup.py	9/2/2018 10:50 AM	PY File	1 KB	
test.json	9/2/2018 11:36 AM	JSON File	14 KB	

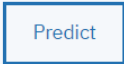
4. Select the contents of the file by placing the cursor to the left of the { and pressing and holding the <Shift><Ctrl><End> keys.

Single Convolution Layer on MNIST Deployed

Overview Implementation Test



Enter input data  

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

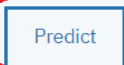


Single Convolution Layer on MNIST Deployed

Overview Implementation Test

Enter input data  

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



6. Based on the confidence levels returned, we can see that the number 7 would be selected as the best fit for this sample image.

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
    ]
  ]
}
```

We Value Your Feedback!

- Don't forget to submit your Think 2019 session and speaker feedback! Your feedback is very important to us – we use it to continually improve the conference.
- Access the Think 2019 agenda tool to quickly submit your surveys from your smartphone, laptop or conference kiosk.

