

## Neural Network Design Overview

This lab will introduce the IBM Flow Modeler for deep learning flows. The lab will consist of the following steps.

1. Creating the data flow (from example)
2. Creating the input data
3. Configuring the input data in the model
4. Run an Experiment on the Training Definition
5. Save and deploy the Model
6. Test the Model

The IBM flow modeler offers a graphical interface for creating deep learning flows. Design deep models for the following types of data: image (CNN architecture), as well as text and audio data (RNN architecture). The neural network designer supports 31 types of layers. Any architecture that can be designed using the combination of these 31 layers, can be designed by using the flow modeler and then publish it as a training definition file.

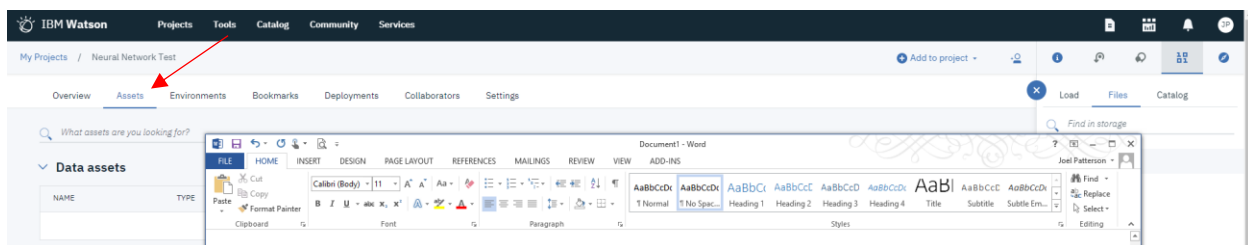
From Wikipedia:

The **CIFAR-10 dataset** ([Canadian Institute For Advanced Research](#)) is a collection of images that are commonly used to train [machine learning](#) and [computer vision](#) algorithms. It is one of the most widely used datasets for machine learning research.<sup>[1][2]</sup> The CIFAR-10 dataset contains 60,000 32x32 color images in 10 different classes.<sup>[3]</sup> The 10 different classes represent airplanes, cars, birds, cats, deer, dogs, frogs, horses, ships, and trucks. There are 6,000 images of each class.<sup>[4]</sup> Computer algorithms for recognizing objects in photos often learn by example. CIFAR-10 is a set of images that can be used to teach a computer how to recognize objects. Since the images in CIFAR-10 are low-resolution (32x32), this dataset can allow researchers to quickly try different algorithms to see what works. Various kinds of [convolutional neural networks](#) tend to be the best at recognizing the images in CIFAR-10.

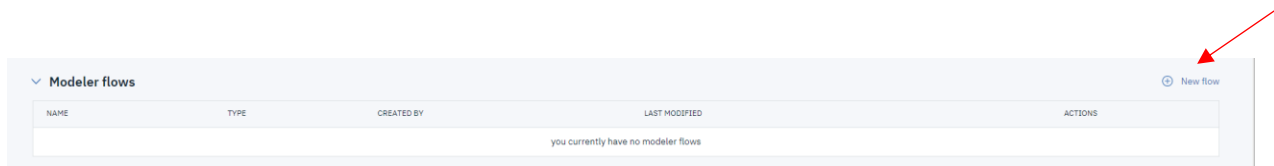
CIFAR-10 is a labeled subset of the 80 million tiny images dataset. When the dataset was created, students were paid to label all of the images.<sup>[5]</sup>

### Step 1: Creating the data flow

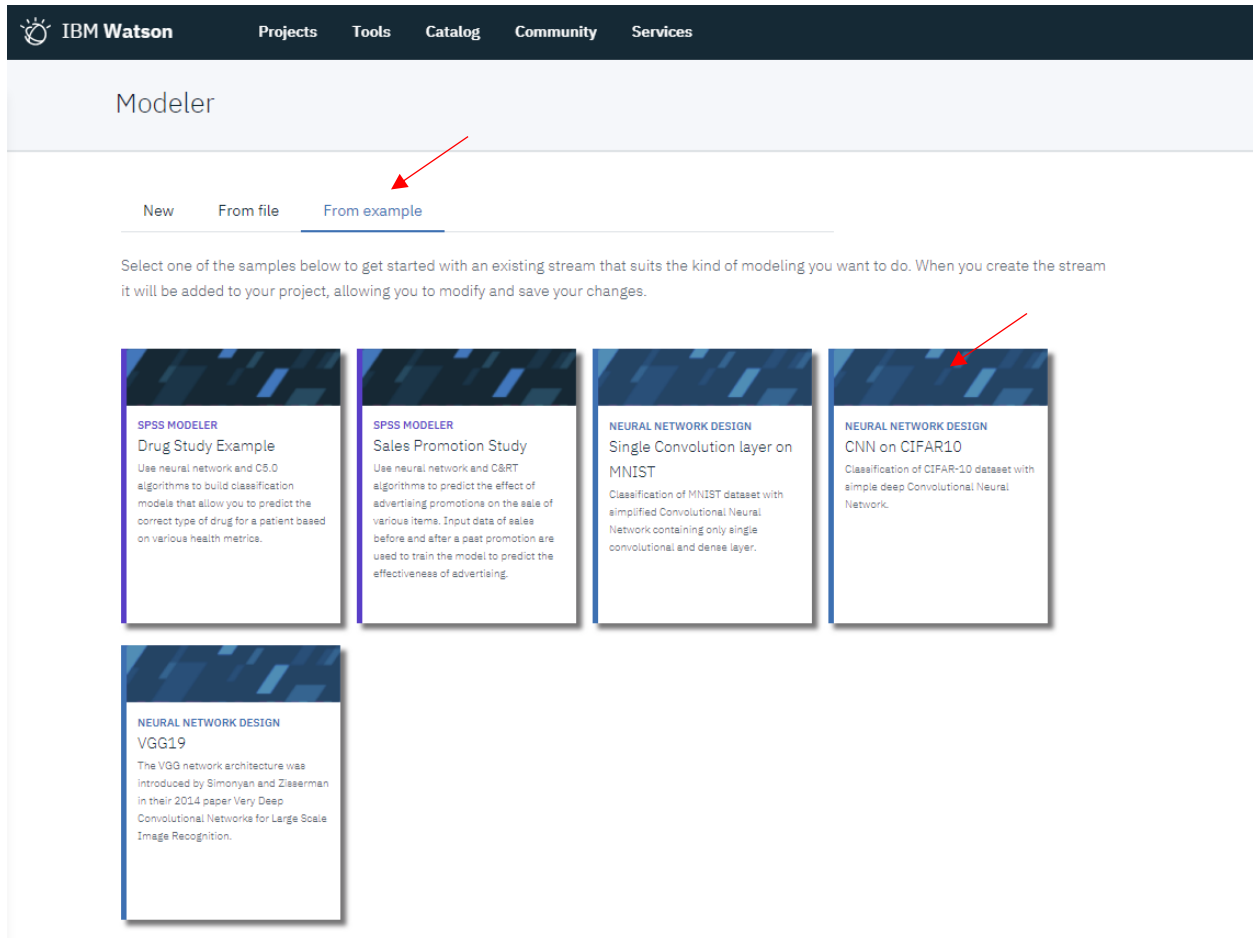
1. Select the **Assets** page of your project



2. Scroll down to **Modeler Flows** and select **New Flow**



### 3. Select **From Example** and select **CNN on CIFAR10**

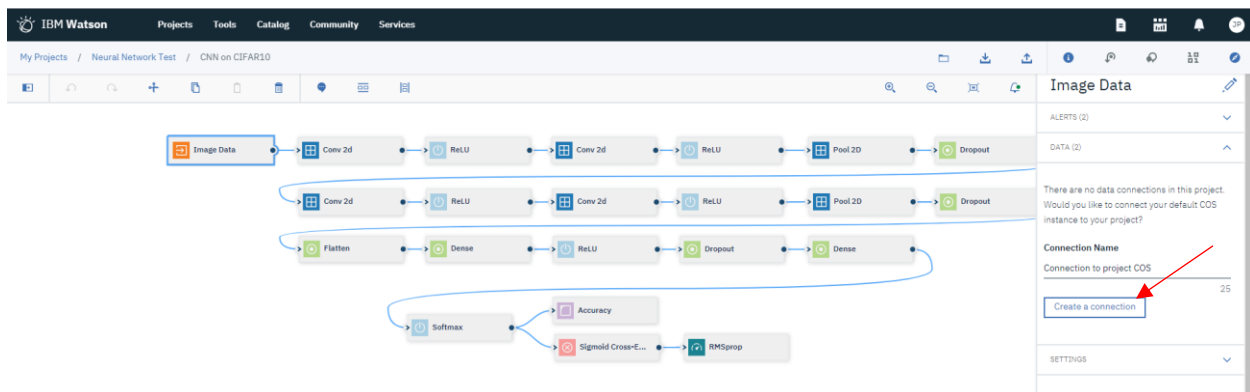


### 4. Select **Create**

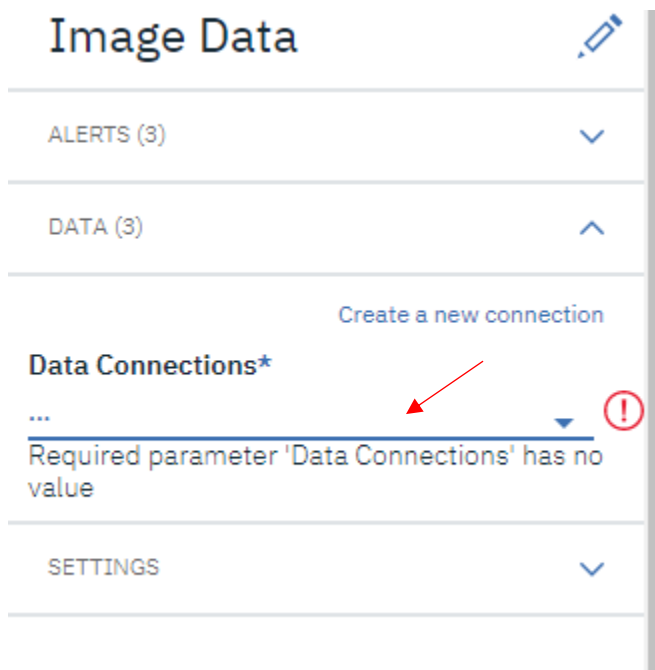
### 5. When the designer appears, double-click the **Image Data** node



### 6. Select **Create a connection** in the **Image Data** configuration panel



7. A data connection will now be created and you should now have the option to set it. Select the **Data Connections** dropdown and set it to **Connection to project COS**.



8. We have not yet created the buckets for the input data, so **Save** this configuration

## Image Data



ALERTS (2)



DATA (2)



Create a new connection

### Data Connections\*

Connection to project COS



### Buckets\*

...



Required parameter 'Buckets' has no value

SETTINGS

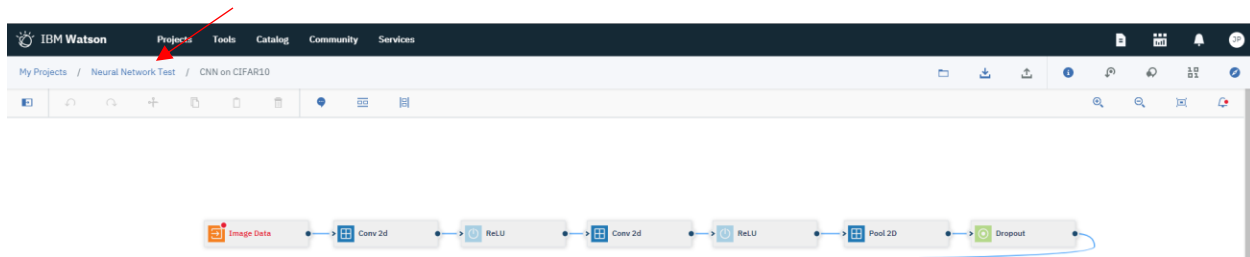


Cancel

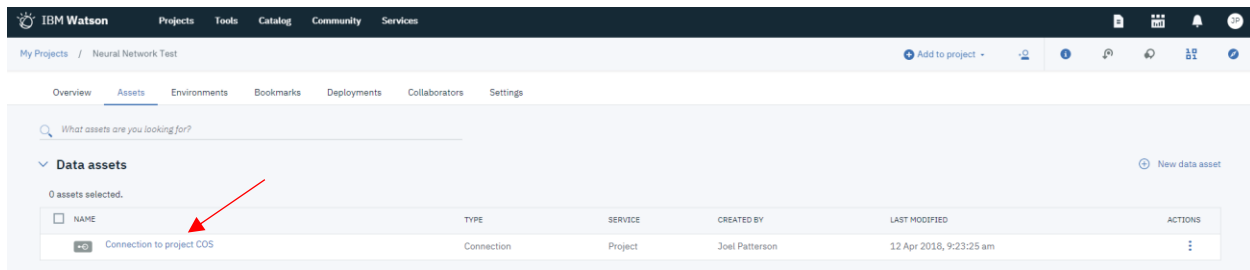
Save

## Step 2: Creating the input data

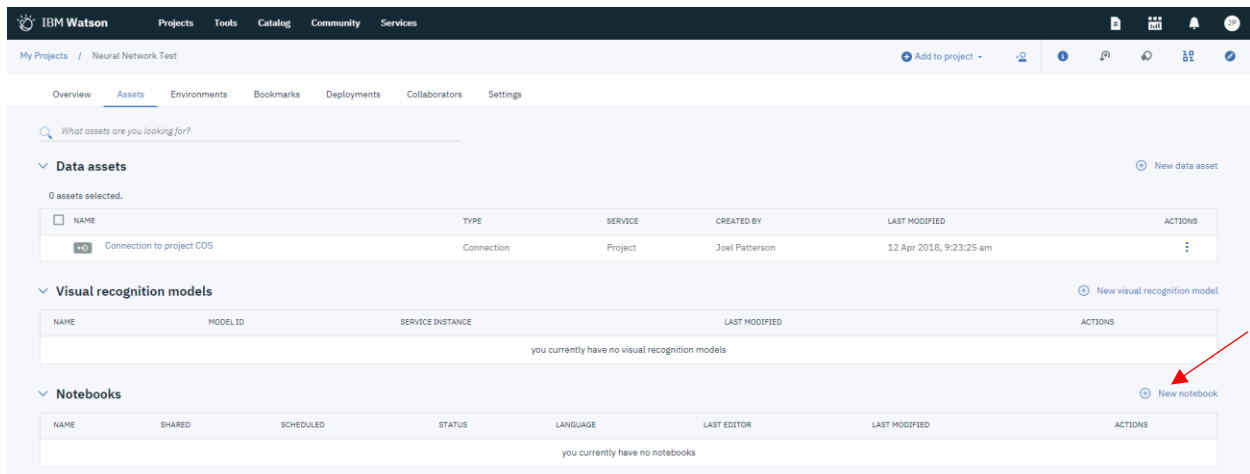
1. Return to the project by selecting the project name in the breadcrumb trail.



2. This returns you to the Assets page. You should see the connection to Cloud Object Storage in the Data Assets section.



3. Add a Notebook asset



4. Select **From URL** to create the notebook from URL. Give the notebook a name (e.g Create Input Data) and optionally a description. Use [https://github.com/bleonardb3/ML-POT/blob/master/Lab-3/CIFAR10\\_data\\_prep.ipynb](https://github.com/bleonardb3/ML-POT/blob/master/Lab-3/CIFAR10_data_prep.ipynb) for the Notebook url. The runtime can remain the **Default Anaconda Free** Environment because we are not using Spark here [note: only one free environment can be running at a time]. Select **Create Notebook**.

IBM Watson Projects Tools Catalog Community Services

My Projects / Neural Network Test / Add Notebook

### New notebook

Blank From file **From URL**

Name\*  
CIFAR10\_data\_prep 33 Characters Remaining

Description  
Type your Description here

Notebook URL\*  
https://github.com/bleonardb3/DSX/blob/master/Lab-1/databaseConnectAndInteract.ipynb

Select runtime\* Includes notebook environments ⓘ  
Default Anaconda Free (1vCPU and 4GB RAM) ✓

The selected Runtime has 1vCPU and 4GB RAM and is free.  
[Learn more about capacity unit hours and Watson Studio pricing plans.](#)

Cancel Create Notebook

5. Run each cell in the notebook. You will need to insert the credentials to your COS store.

IBM Watson Projects Tools Catalog Community Services

My Projects / Neural Network Test / CIFAR10\_data\_prep

File Edit View Insert Cell Kernel Help

Not Trusted | Python 3.5

```

pickle.dump(X_test, y_test), f, protocol=pickle.HIGHEST_PROTOCOL)

In [ ]: import boto3
        from ibm_botocore.client import Config
  
```

**Insert Cloud Object Storage credentials**

1. Select the Data icon (upper right, icon with 1 and 0s)
2. Select Connections
3. Select the cell below and then select **Insert to code** for **Connection to project COS** to insert credentials
4. Rename the credentials (which are usually called credentials\_1) to cos\_credentials. YOU WILL GET ERRORS IF YOU DO NOT DO THIS!

In [ ]:

Connections

Connection to project COS  
Insert to code

6. You should see something similar to the following after running the cells.

**Create bucket name -- I usually append my name to the end (i.e. cifar10-tutorials-joel)**

```
In [10]: bucket = "cifar10-tutorials-test"
```

**This call may fail if bucket has already been created -- that's fine, simply continue executing the next cells**

```
In [11]: cos.create_bucket(Bucket=bucket)
```

```
Out[11]: {'ResponseMetadata': {'HTTPHeaders': {'content-length': '0',
'date': 'Thu, 12 Apr 2018 14:56:13 GMT',
'server': 'Cleversafe/3.12.4.32',
'x-amz-request-id': 'b10420b3-e211-4d5b-9081-f35962e7a4cd',
'x-clv-request-id': 'b10420b3-e211-4d5b-9081-f35962e7a4cd',
'x-clv-s3-version': '2.5'},
'HTTPStatusCode': 200,
'HostId': '',
'RequestId': 'b10420b3-e211-4d5b-9081-f35962e7a4cd',
'RetryAttempts': 0}}
```

```
In [12]: files = ['cifar-10-tf-train.pkl',
'cifar-10-tf-valid.pkl',
'cifar-10-tf-test.pkl']
```

```
In [13]: for file in files:
print('Uploading data {}'.format(file))
cos.upload_file(Filename=file, Bucket=bucket, Key=file)
```

Uploading data {}... cifar-10-tf-train.pkl  
Uploading data {}... cifar-10-tf-valid.pkl  
Uploading data {}... cifar-10-tf-test.pkl

```
In [14]: cos.list_objects(Bucket=bucket)
```

```
Out[14]: {'Contents': [{'ETag': '"86e585d89b77f29ea770732cf8cd834a-4"',
'Key': 'cifar-10-tf-test.pkl',
'LastModified': datetime.datetime(2018, 4, 12, 14, 56, 28, 880000, tzinfo=tzlocal()),
'Owner': {'DisplayName': '5745ac3f-6ddc-427f-9e6f-733419d40aa3',
'ID': '5745ac3f-6ddc-427f-9e6f-733419d40aa3'},
'Size': 30800251,
'StorageClass': 'STANDARD'}]}
```

7. The ending cells are for getting data to test the deployed model. We'll come back to this.

**The number of the image we wish to test from the validation set**

```
In [15]: image_number = 110
```

```
In [16]: with open("cifar-10-tf-valid.pkl", "rb") as f:
cifar_test_data=pickle.load(f)
cifar_test_data[0][image_number].tolist()
```

```
Out[16]: [[170, 196, 133],
[109, 210, 150],
[132, 150, 100],
[120, 146, 95],
[120, 145, 90],
[70, 80, 40],
[59, 71, 41],
[40, 63, 40],
[61, 77, 55],
[57, 78, 53],
[66, 93, 66],
[70, 107, 90],
[40, 59, 42],
[40, 64, 46],
[112, 130, 104],
[106, 126, 92],
[102, 125, 92],
[104, 130, 99],
[103, 120, 99],
[103, 120, 99]]
```

```
In [17]: print (cifar_test_data[3][110])
```

```
[6]
```

## Step 3: Configuring the input data in the model

1. Return to the project by selecting the project name in the breadcrumb trail

The number of the image we wish to test from the validation set

```
In [15]: image_number = 110

In [16]: with open("cifar-10-tf-valid.pkl", "rb") as f:
cifar_test_data=pickle.load(f)
cifar_test_data[0][image_number].tolist()

Out[16]: [[170, 190, 135],
[109, 210, 156],
[132, 150, 100],
[120, 146, 95],
[120, 145, 90],
[70, 80, 40],
[59, 71, 41],
[40, 63, 40],
[61, 77, 55],
[57, 78, 53],
[66, 93, 66],
[70, 107, 90],
[40, 59, 42],
[40, 64, 46],
[112, 130, 104],
[100, 120, 92],
[102, 125, 92],
[104, 130, 99],
[109, 120, 99],
[109, 120, 99]]

In [17]: print (cifar_test_data[1][110])

[6]
```

2. You should be in the **Assets** tab. Scroll down to **Modeler Flows** and open the **CNN** flow

Modeler flows					New flow
NAME	TYPE	CREATED BY	LAST MODIFIED	ACTIONS	
CNN on CIFAR10	Neural Network	Joel Patterson	12 Apr 2018, 9:58:50 am		

3. Double-click the **Image Data** icon to open the configuration window.



4. Set **Buckets** to the value of the bucket you just created in the notebook.



# Image Data

ALERTS (1)

DATA (1)

Create a new connection

**Data Connections\***  
Connection to project COS

**Buckets\***  
cifar10-tutorials-test

Select files from your bucket to specify train, validation and test data.

**Train data file\***  
...  
Required parameter 'Train data file' has no value

**Test data file**  
...

**Validation data file**  
...

5. Set the **Train, Test, Validate** files to the values in the bucket (you wrote these out in the notebook). Then **Save**.

# Image Data



DATA



Create a new connection

## Data Connections\*

Connection to project COS



## Buckets\*

cifar10-tutorials-test



Select files from your bucket to specify train, validation and test data.

## Train data file\*

cifar-10-tf-train.pkl



## Test data file

cifar-10-tf-test.pkl



## Validation data file

cifar-10-tf-valid.pkl



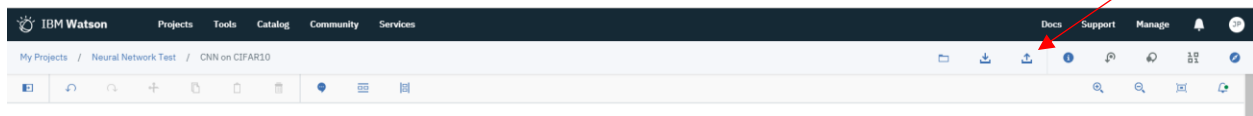
SETTINGS



Cancel

Save

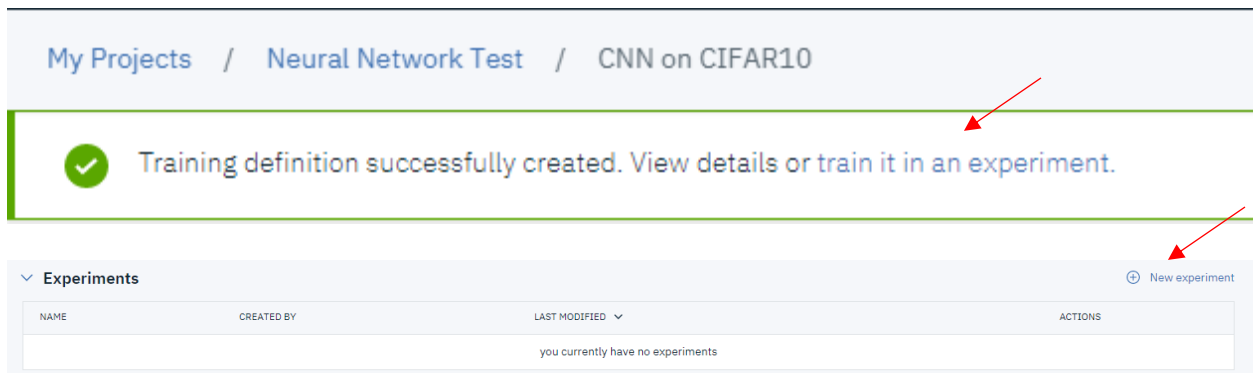
## 6. Publish Training Definition (container with up-pointing arrow)



7. Enter a **Name** for the definition and pick which WML Instance it should be published to (this is in case you have more than one defined). Select **Publish**.


A screenshot of the 'Publish Training Definition' form. The form has a title 'Publish Training Definition' and a subtitle 'Fill name and description then select WML instance to publish.' Below the subtitle, there are three main sections: 1. 'Name\*' with a text input field containing 'CNN on CIFAR10' and a character count of 36. 2. 'Description' with a text area containing 'Optional Description' and a character count of 200. 3. 'Select WML Instance\*' with a dropdown menu showing 'WML'. At the bottom right of the form, there are two buttons: 'Cancel' and 'Publish'. A red arrow points to the 'Publish' button.

8. Upon successfully publication you have the option to train the definition in an experiment. You can select the quick link here or via the **Experiments** section in **Assets**.



## Step 4: Run an Experiment on the Training Definition


1. Give a **Name** to the Experiment and choose a **Machine Learning Service**

 **IBM Watson** [Projects](#) [Tools](#) [Catalog](#) [Community](#) [Services](#)

# New experiment BETA

### Define experiment details

**Name**

CNN on CIFAR10 Experiment| 


75

**Description**

Experiment description

300

**Machine Learning Service**

WML 

▼

**Cloud Object Storage bucket for storing training source and results files**

Select

2. Click the **Select** link for **Cloud Object Storage bucket**



## New experiment <sup>BETA</sup>

### Define experiment details

#### Name

CNN on CIFAR10 Experiment|

75

#### Description

*Experiment description*

300

#### Machine Learning Service

WML



Cloud Object Storage bucket for storing training source and results files

Select



3. Using **Existing connections**, select your **Cloud Object Storage connection** and set both buckets to the bucket you created in the notebook. Click **Select**.

IBM Watson
Projects
Tools
Catalog
Community
Services
Docs
Support
Manage
JP

## Cloud Object Storage bucket selection

Existing connections

New connection

Cloud Object Storage connection

Connection to project COS

Bucket containing training data

☒ Existing
 ☐ New

cifar10-tutorials-test

Bucket for storing training results

☒ Existing
 ☐ New

cifar10-tutorials-test

Cancel

Select

#### 4. Select **Add training definition**

IBM Watson
Projects
Tools
Catalog
Community
Services

## New experiment BETA

Define experiment details

Name

CNN on CIFAR10 Experiment

75

Description

Experiment description

300

Machine Learning Service

WML

Cloud Object Storage buckets for storing training source and results files

Source: Connection to project COS / cifar10-tutorials...

Results: Connection to project COS / cifar10-tutorials...

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
No training definitions associated.	

☐ Use global execution command (override training definition values)

#### 5. Select **Existing training definition** and choose the definition you saved.

 IBM Watson

ProjectsToolsCatalogCommunityServices

Add training definition

New training definitionExisting training definition

Select training definition

Existing training definitions

CNN on CIFAR10

6. Select **Training definition attributes**. **Compute plan** should be ½ x NVIDIA Tesla K80 (1 GPU) – free accounts are only allowed this selection. **Hyperparameter optimization method** can be any value, but **none** will run the quickest. Click **Select**.

Training definition attributes

Compute plan

1/2 x NVIDIA® Tesla® K80 (1 GPU)

Hyperparameter optimization method

none

CancelSelect

7. Select **Create and run**

[Projects](#)
[Tools](#)
[Catalog](#)
[Community](#)
[Services](#)

[Docs](#)
[Support](#)
[Manage](#)

New experiment BETA

Define experiment details

Name

CNN on CIFAR10 Experiment

75

Description

Experiment description

300

Machine Learning Service

WML

▼

Cloud Object Storage buckets for storing training source and results files

Source: [Connection to project COS / cifar10-tutorials...](#)

Results: [Connection to project COS / cifar10-tutorials...](#) [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
CNN on CIFAR10	1/2 x NVIDIA® Tesla® K80 (1 GPU)

☐ Use global execution command (override training definition values)

Cancel

Create and run

- Training neural networks is resource intensive. Therefore, this experiment will take an hour or so to complete. Start working on the next lab and we will come back to this later.

[Projects](#)
[Tools](#)
[Catalog](#)
[Community](#)
[Services](#)

[Docs](#)
[Support](#)
[Manage](#)

My Projects / Neural Network Test / CNN on CIFAR10 Experiment

CNN on CIFAR10 Experiment

[Cancel runs in progress](#)
[Add training runs](#)

Training Runs

Compare Runs

Overview

1

Runs in total

0 hr, 2 min, 3 sec

Total running time

Queued

NAME	SUBMITTED
No training runs found.	

In progress

NAME	DURATION
CNN on CIFAR10	0 hr, 2 min, 3 sec

Completed

NAME	STATUS	DURATION	ACTIONS
No training runs found.			



IBM Watson Projects Tools Catalog Community Services Docs Support Manage

My Projects / Neural Network Test / CNN on CIFAR10 Experiment

### CNN on CIFAR10 Experiment

Cancel runs in progress Add training runs

Training Runs Compare Runs Overview

1 Runs in total 1 hr, 14 min, 26 sec Total running time

#### Queued

NAME	SUBMITTED
No training runs found.	

#### In progress

NAME	DURATION
No training runs found.	

#### Completed

NAME	STATUS	DURATION	ACTIONS
CNN on CIFAR10	completed	1 hr, 14 min, 26 sec	

## Step 5: Save and deploy the Model

### 1. Select the completed training run

IBM Watson Projects Tools Catalog Community Services Docs Support Manage

My Projects / Neural Network Test / CNN on CIFAR10 Experiment

### CNN on CIFAR10 Experiment

Cancel runs in progress Add training runs

Training Runs Compare Runs Overview

1 Runs in total 1 hr, 14 min, 26 sec Total running time

#### Queued

NAME	SUBMITTED
No training runs found.	

#### In progress

NAME	DURATION
No training runs found.	

#### Completed

NAME	STATUS	DURATION	ACTIONS
CNN on CIFAR10	completed	1 hr, 14 min, 26 sec	

### 2. View Overview and Logs

IBM Watson Projects Tools Catalog Community Services Docs Support Manage

My Projects / Neural Network Test / CNN on CIFAR10 Experiment / CNN on CIFAR10

### CNN on CIFAR10

Monitor Overview **Logs**

```

> 9440/10000 [=====] - ETA: 0s
> 9696/10000 [=====] - ETA: 0s
> 9952/10000 [=====] - ETA: 0s
> 10000/10000 [=====] - 2s 219us/step
> /usr/local/lib/python3.5/dist-packages/h5py/_init_.py:36: FutureWarning: Conversion of the second argument of issubdtype from 'float' to 'np.floating' is deprecated. In future, it will be treated as 'np.float64 == np.dtype(float).type'.
> from _conv import register_converters as _register_converters
> Using TensorFlow backend.
> [0.7742609910011291, 0.7375]
> Saving the model...
> Model saved in file: /mnt/results/cifar10-tutorials-test/training-sBmGfOWIR/model/CNN-on-CIFAR10.h5
> Done!
> .....
  
```

Most recent 500 logs are displayed.

3. Return to the Experiment by selecting it in the breadcrumb trail

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

My Projects / Neural Network Test / CNN on CIFAR10 Experiment / CNN on CIFAR10

### CNN on CIFAR10

Monitor Overview **Logs**

```

> 9440/10000 [=====] - ETA: 0s
> 9696/10000 [=====] - ETA: 0s
> 9952/10000 [=====] - ETA: 0s
> 10000/10000 [=====] - 2s 219us/step
> /usr/local/lib/python3.5/dist-packages/h5py/_init_.py:36: FutureWarning: Conversion of the second argument of issubdtype from 'float' to 'np.floating' is deprecated. In future, it will be treated as 'np.float64 == np.dtype(float).type'.
> from _conv import register_converters as _register_converters
> Using TensorFlow backend.
> [0.7742609910011291, 0.7375]
> Saving the model...
> Model saved in file: /mnt/results/cifar10-tutorials-test/training-sBmGfOWIR/model/CNN-on-CIFAR10.h5
> Done!
> .....
  
```

Most recent 500 logs are displayed.

4. Select the **Actions** for the completed Experiment and **Save model**

IBM Watson Projects Tools Catalog Community Services Docs Support Manage

My Projects / Neural Network Test / CNN on CIFAR10 Experiment

### CNN on CIFAR10 Experiment

Training Runs Compare Runs Overview

1 Runs in total 1 hr, 14 min, 26 sec Total running time

**Queued**

NAME	SUBMITTED
No training runs found.	

**In progress**

NAME	DURATION
No training runs found.	

**Completed**

NAME	STATUS	DURATION	ACTIONS
CNN on CIFAR10	completed	1 hr, 14 min, 26 sec	Save model

5. Give a **Name** to the model and select **Save**

IBM Watson

ProjectsToolsCatalogCommunityServices

DocsSupportManage

JP

Save Model

Name

CNN of CIFAR10 Model

80

Description

Experiment description

300

Cancel

Save

6. Upon successful save you can view the model by selecting the link or by viewing the model in the **Assets** tab.

IBM Watson

ProjectsToolsCatalogCommunityServices

DocsSupportManage

JP

My Projects / Neural Network Test / CNN on CIFAR10 Experiment

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

Models

New model

NAME	STATUS	TYPE	RUNTIME	LAST MODIFIED	ACTIONS
CNN of CIFAR10 Model	trained	tensorflow-1.5	python-3.5	16 Apr 2018	

7. Select the **Deployments** tab

IBM Watson Projects Tools Catalog Community Services Docs Support Manage JP

My Projects / Neural Network Test / CNN of CIFAR10 Model

### CNN of CIFAR10 Model

Overview Evaluation **Deployments**

#### Summary

Machine learning service	WML
Model Type	tensorflow-1.5
Runtime environment	python-3.5
Training date	16 Apr 2018, 1:25 PM
Latest version	09ce4cc9-0045-45c0-93d2-d4b8ebbe49fa

## 8. Select **Add deployment**

IBM Watson Projects Tools Catalog Community Services Docs Support Manage JP

My Projects / Neural Network Test / CNN of CIFAR10 Model

### CNN of CIFAR10 Model

Overview Evaluation **Deployments**

Add Deployment

NAME	STATUS	DEPLOYMENT TYPE	ACTIONS
Your model is not deployed.			

## 9. Give a **Name** to the deployment and select **Save**

IBM Watson

ProjectsToolsCatalogCommunityServices

DocsSupportManage

JP

Create Deployment

Web ServiceBatch Prediction

Name

CNN on CIFAR10 Deployment

Description

Web Service Deployment Description

300

Cancel

Save

10. Wait until deployment is successful (refreshing the page may help here)

IBM Watson

ProjectsToolsCatalogCommunityServices

DocsSupportManage

My Projects / Neural Network Test / CNN of CIFAR10 Model

CNN of CIFAR10 Model

OverviewEvaluationDeployments

Add Deployment

NAME	STATUS	DEPLOYMENT TYPE	ACTIONS
CNN on CIFAR10 Deployment	DEPLOY_SUCCESS	Web Service	

## Step 6: Test the Model

1. Select the deployed model

IBM Watson

ProjectsToolsCatalogCommunityServices

DocsSupportManage

My Projects / Neural Network Test / CNN of CIFAR10 Model

CNN of CIFAR10 Model

OverviewEvaluationDeployments

Add Deployment

NAME	STATUS	DEPLOYMENT TYPE	ACTIONS
CNN on CIFAR10 Deployment	DEPLOY_SUCCESS	Web Service	

2. Select **Test** to test the model

## CNN on CIFAR10 Deployment

[Overview](#)

[Implementation](#)

[Test](#)

### Enter input data



*Paste the request payload here*

Predict

3. Right-click on the project breadcrumb and **Open in new tab**

## CNN on CIFAR10 Deployment

OverviewImplementationTest

### Enter input data



Paste the request payload here

Predict

4. Open the CIFAR10 notebook you created earlier.

IBM Watson Projects Tools Catalog Community Services

My Projects / Neural Network Test

Overview Assets Environments Bookmarks Deployments Collaborators Settings

What assets are you looking for?

▼ Data assets

0 assets selected.

NAME	TYPE	SERVICE	CREATED BY	LAST MODIFIED	ACTIONS
Connection to project COS	Connection	Project	Joel Patterson	12 Apr 2018, 9:23:25 am	

▼ Visual recognition models

you currently have no visual recognition models

▼ Notebooks

NAME	SHARED	SCHEDULED	STATUS	LANGUAGE	LAST EDITOR	LAST MODIFIED	ACTIONS
CIFAR10_data_prep				Python 3.5	Joel Patterson	16 Apr 2018	

5. Copy the image data at the end of the notebook. Also take note of the value at the end.

IBM Watson Projects Tools Catalog Community Services

My Projects / Neural Network Test / CIFAR10\_data\_prep

```
[85, 87, 71],
[92, 98, 81],
[77, 86, 68],
[72, 84, 65],
[76, 87, 71],
[81, 87, 80],
[79, 82, 78],
[67, 68, 59],
[64, 69, 59],
[62, 70, 60],
[62, 71, 57],
[20, 27, 14],
[25, 31, 19],
[29, 35, 24],
[32, 35, 24],
[39, 41, 30],
[24, 26, 15],
[23, 25, 15],
[53, 57, 37],
[80, 89, 54],
[54, 63, 35],
[45, 52, 34],
[34, 30, 29],
[40, 42, 40],
[50, 50, 51],
[46, 46, 48],
[81, 81, 84],
[84, 83, 83],
[86, 86, 79],
[81, 81, 73],
[95, 95, 88],
[81, 81, 74],
[79, 80, 67],
[77, 81, 67],
[88, 94, 79],
[76, 84, 69],
[69, 78, 65],
[89, 95, 89],
[107, 110, 106],
[98, 100, 90],
[80, 87, 76],
[78, 88, 79],
[87, 98, 84]]
```

```
In [17]: print (cifar_test_data[1][110])
[6]
```

6. Paste this into the test window.





## CNN on CIFAR10 Deployment

Overview

Implementation

Test

### Enter input data



```
[[[170, 196, 135],  
  [189, 210, 156],  
  [132, 150, 100],  
  [128, 146, 95],  
  [126, 145, 96],  
  [70, 88, 48],  
  [53, 71, 41],  
  [46, 63, 40],  
  [64, 88, 55]]]
```

Predict

7. Add { "values": [ to the top



## CNN on CIFAR10 Deployment

[Overview](#)

[Implementation](#)

[Test](#)

### Enter input data



```
{  
  "values": [  
    [[170, 196, 135],  
     [189, 210, 156],  
     [132, 150, 100],  
     [128, 146, 95],  
     [126, 145, 96],  
     [70, 88, 48],  
     [58, 84, 141]
```

Predict

8. Add `]]` to the bottom



## CNN on CIFAR10 Deployment

[Overview](#)

[Implementation](#)

[Test](#)

### Enter input data



```
[89, 95, 89],  
[107, 110, 106],  
[98, 100, 90],  
[80, 87, 76],  
[78, 88, 79],  
[87, 98, 84]]]  
}  
}
```



Predict

9. **Predict.** You should get an array of 10 values [0-9 scale]. The maximum value should map to the index value you saw in the notebook.



## CNN on CIFAR10 Deployment

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

### Enter input data



```
[89, 95, 89],  
[107, 110, 106],  
[98, 100, 90],  
[80, 87, 76],  
[78, 88, 79],  
[87, 98, 84]]]  
]  
}
```

Predict

```
{  
  "fields": [  
    "prediction"  
  ],  
  "values": [  
    [  
      2.917013262049295e-7,  
      0.0000013955738040749566,  
      0.00018943203031085432,  
      0.0011891975300386548,  
      0.00040439344593323767,  
      0.000057278768508695066,  
      0.9981558918952942,  
      0.0000018143621218769113,  
      1.1386251941303271e-7,  
      1.83896190719679e-7  
    ]  
  ]  
}
```