

SageMaker with DGL integration

Example 1 - Semi-supervised nodes classification

At first, new SageMaker service account needs to be created. SageMaker must be searched within AWS services.

Get started

Quick start

Let Amazon SageMaker handle configuring account and setting the permissions that you or a team in your organization need to use Amazon SageMaker Studio. Choosing this option uses standard encryption, which you can't change. If you need more control over configuration, choose Standard setup.

User name

sage-maker-dgl-test

The user name can have up to 63 characters. Valid characters: A-Z, a-z, 0-9, and - (hyphen)

Execution role

Amazon SageMaker Studio requires permissions to access other AWS services, such as Amazon SageMaker and Amazon S3. The execution role must have the [AmazonSageMakerFullAccess policy](#) attached. If you don't have a role with this policy attached, we can create one for you.

Create a new role

Standard setup

Control all aspects of account configuration, including permissions and encryption. Choose this option if you are an administrator setting up Amazon SageMaker Studio for your organization.

Zrob przut ekranu

Cancel

Submit

To make use of majority of SageMaker features, S3 bucket is needed. The UI creator helps to create one on demand:

Create an IAM role

Passing an IAM role gives Amazon SageMaker permission to perform actions in other AWS services on your behalf. Creating a role here will grant permissions described by the [AmazonSageMakerFullAccess](#) IAM policy to the role you create.

The IAM role you create will provide access to:

S3 buckets you specify - optional

Any S3 bucket

Allow users that have access to your notebook instance access to any bucket and its contents in your account.

Specific S3 buckets

Example: bucket-name-1, bucket-name-2

Comma delimited. ARNs, "*" and "/" are not supported.

None

Any S3 bucket with "sagemaker" in the name

Any S3 object with "sagemaker" in the name

Any S3 object with the tag "sagemaker" and value "true"

[See Object tagging](#)

S3 bucket with a Bucket Policy allowing access to SageMaker

[See S3 bucket policies](#)

Cancel

Create role

Execution role

Amazon SageMaker Studio requires permissions to access other AWS services, such as Amazon SageMaker and Amazon S3. The execution role must have the [AmazonSageMakerFullAccess policy](#) attached. If you don't have a role with this policy attached, we can create one for you.

AmazonSageMaker-ExecutionRole-20200628T195130

Success! You created an IAM role.

AmazonSageMaker-ExecutionRole-20200628T195130

After initialization is completed - one can access SageMaker Studio Control Panel

Amazon SageMaker Studio is ready
Choose your user name, then choose Open Studio to get started.

Amazon SageMaker > Amazon SageMaker Studio > Control Panel

Amazon SageMaker Studio Control Panel

Choose your user name, then choose Open Studio to get started


< 1 > ⚙

User name	Last modified	Created	
sage-maker-dgl-test	Jun 28, 2020 19:55 UTC	Jun 28, 2020 19:54 UTC	Open Studio ↗

▼ Studio Summary [How to delete Studio](#) [Delete Studio](#)

Status Ready	Studio ID d-b0yxcbbbus	Execution role arn:aws:iam::230316164119:role/service-role/AmazonSageMaker-ExecutionRole-20200628T195130	Authentication method AWS Identity and Access Management (IAM)
-----------------	---------------------------	---	---

With use of **Open Studio** button one can initialize SageMaker Studio which is kind of a IDE based on Jupyter to train and deploy ML models.



Amazon SageMaker Studio

Loading the JupyterServer application default...

Amazon SageMaker Studio

File Edit View Run Kernel Git Tabs Settings Help

Launcher Amazon SageMaker Studio

Welcome to Amazon SageMaker Studio

Build and train

Spin up sharable Jupyter Notebooks in seconds to build ML models and launch new experiments. Easily organize, track and compare experiments using SageMaker Experiments. Run distributed training, and troubleshoot models with SageMaker Debugger.

[Create a notebook](#)

Deploy and monitor

Deploy your models with auto scaling, and automatically monitor for drift in production using SageMaker Model Monitor.

[Deploy and monitor models](#)

Build models automatically

Automatically build, train, and tune models with full visibility and control, using SageMaker Autopilot.

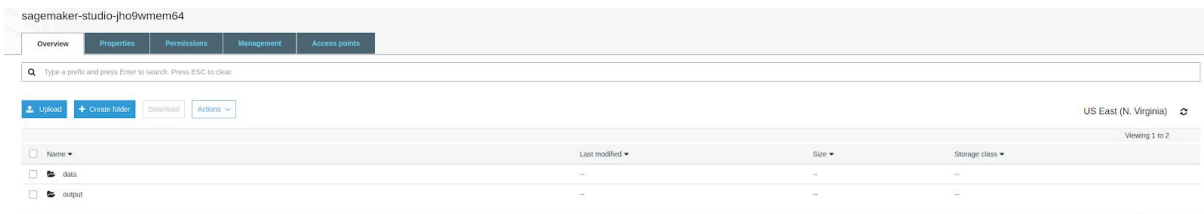
[Create Autopilot experiment](#)

Learn more

Watch video tutorials [↗](#) to learn more about SageMaker Studio.
Read the end-to-end Studio [tour guide](#) [↗](#) with examples.

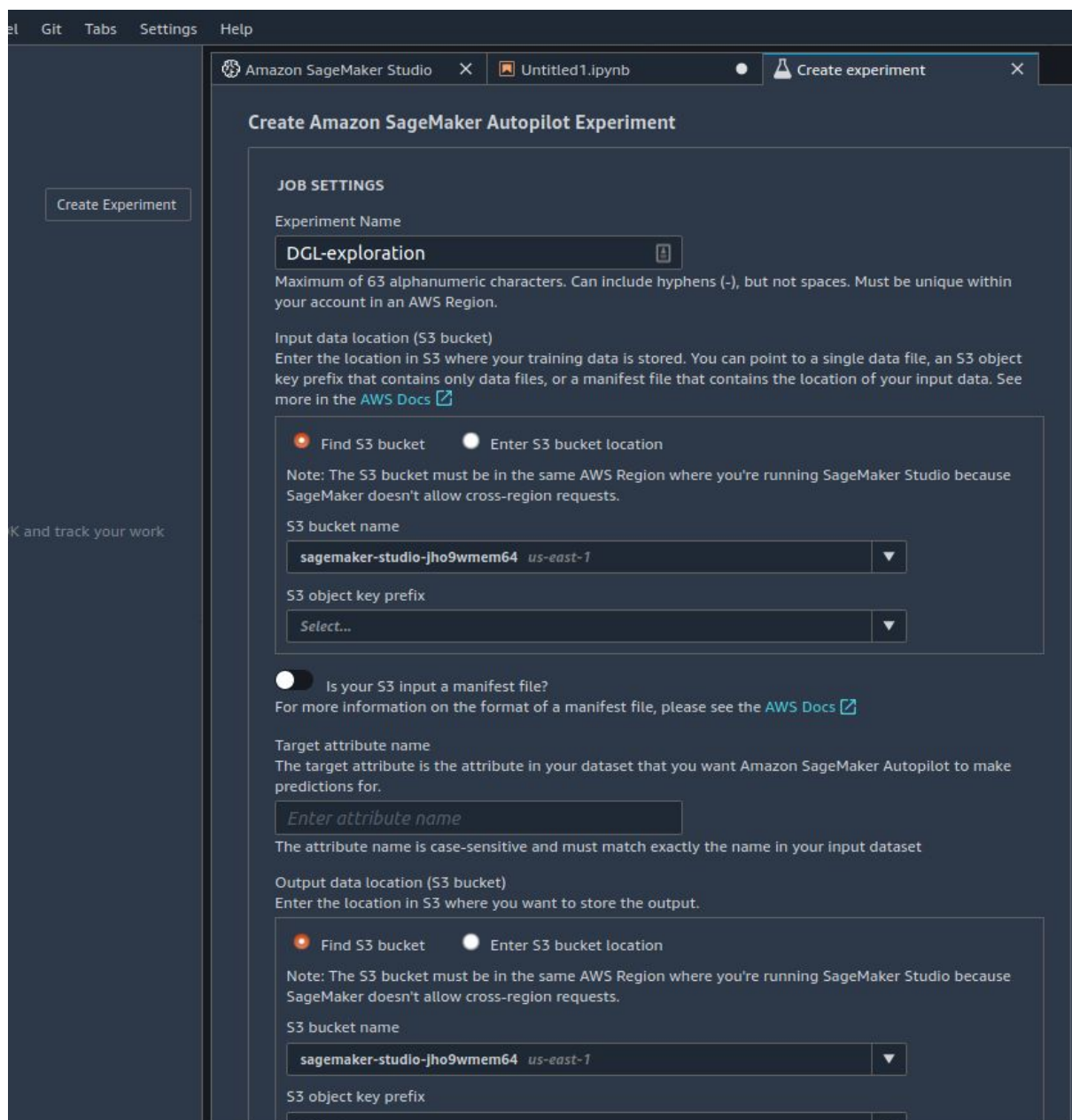
Version: 1.20.1

To create an experiment (**Create Experiment** button) one needs some structure on S3 bucket (to store input and output data in controllable manner). Here the data *directory* contains training dataset.



Name	Last modified	Size	Storage class
data	--	--	--
output	--	--	--

One need to fulfill the form with proper values regarding data structure.



Create Amazon SageMaker Autopilot Experiment

JOB SETTINGS

Experiment Name
DGL-exploration

Maximum of 63 alphanumeric characters. Can include hyphens (-), but not spaces. Must be unique within your account in an AWS Region.

Input data location (S3 bucket)
Enter the location in S3 where your training data is stored. You can point to a single data file, an S3 object key prefix that contains only data files, or a manifest file that contains the location of your input data. See more in the [AWS Docs](#)

☒ Find S3 bucket ☐ Enter S3 bucket location

Note: The S3 bucket must be in the same AWS Region where you're running SageMaker Studio because SageMaker doesn't allow cross-region requests.

S3 bucket name
sagemaker-studio-jho9wmem64 us-east-1

S3 object key prefix
Select...

☐ Is your S3 input a manifest file?
For more information on the format of a manifest file, please see the [AWS Docs](#)

Target attribute name
The target attribute is the attribute in your dataset that you want Amazon SageMaker Autopilot to make predictions for.
Enter attribute name

The attribute name is case-sensitive and must match exactly the name in your input dataset

Output data location (S3 bucket)
Enter the location in S3 where you want to store the output.

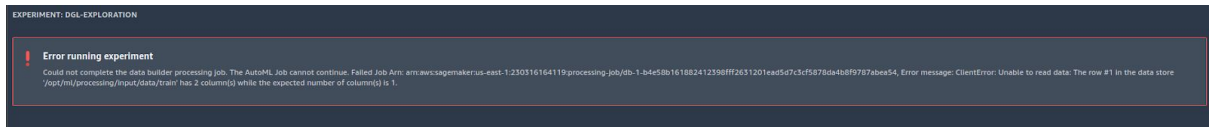
☒ Find S3 bucket ☐ Enter S3 bucket location

Note: The S3 bucket must be in the same AWS Region where you're running SageMaker Studio because SageMaker doesn't allow cross-region requests.

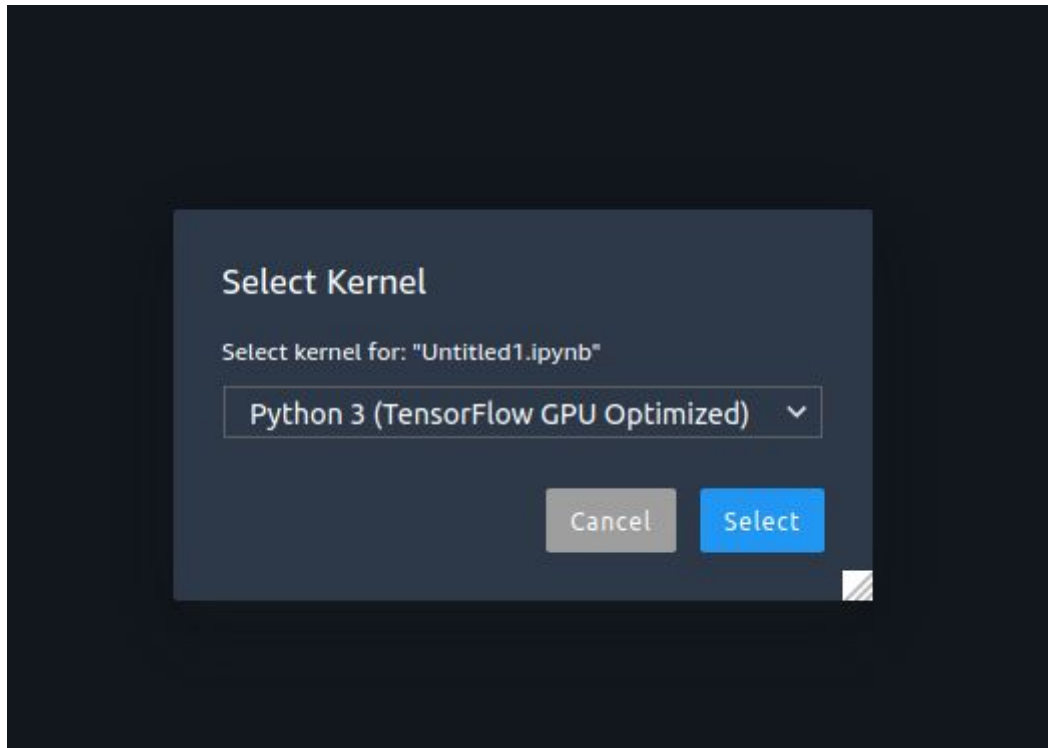
S3 bucket name
sagemaker-studio-jho9wmem64 us-east-1

S3 object key prefix

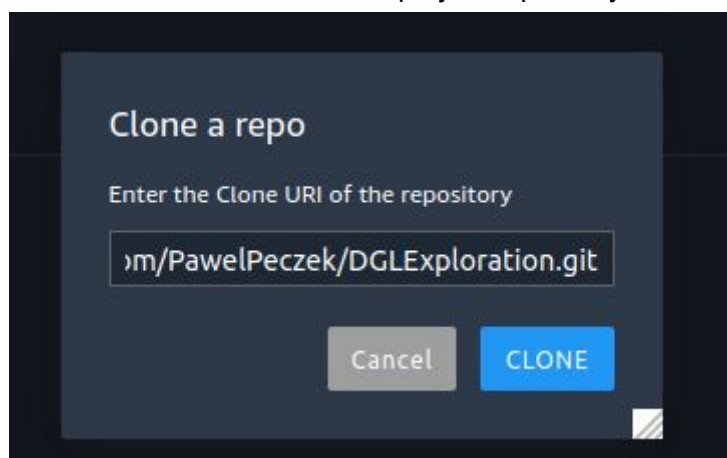
In general case (when handling unstructured data - like in this case), automatic machinery will not be able to help with data indexing.



One can trigger new notebook (and choose runtime env).



To have an access to a code one need to clone project repository



As this is done, data must be downloaded from S3:

```

Name                               Last Modified
test_simulation_snapshot_50.json    seconds ago
Untitled1.ipynb                    seconds ago

[1]: import boto3

[3]: s3 = boto3.client('s3')
     s3.download_file('sagemaker-studio-jho9wmem64', 'data/test_simulation_snapshot_50.json', 'test_simulation_snapshot_50.json')

[ ]:
```

Then, the code may be executed in the exact form prepared in repository (via Jupyter Notebooks).

```

Training

[36]: # Training loop from https://docs.dgl.ai/en/0.4.x/tutorials/basics/1_first.html

import itertools

optimizer = torch.optim.Adam(itertools.chain(net.parameters(), embeddings.parameters()), lr=0.01)
all_logits = []
for epoch in range(80):
    logits = net(graph, inputs)
    all_logits.append(logits.detach())
    logp = F.log_softmax(logits, 1)
    loss = F.nll_loss(logp[Labeled_nodes], labels)
    optimizer.zero_grad()
    loss.backward()
    optimizer.step()
    print('Epoch %d | Loss: %.4f' % (epoch, loss.item()))

Epoch 0 | Loss: 0.7092
Epoch 1 | Loss: 0.6650
Epoch 2 | Loss: 0.6287
Epoch 3 | Loss: 0.5960
Epoch 4 | Loss: 0.5661
Epoch 5 | Loss: 0.5398
Epoch 6 | Loss: 0.5142
Epoch 7 | Loss: 0.4900
Epoch 8 | Loss: 0.4663
Epoch 9 | Loss: 0.4425
Epoch 10 | Loss: 0.4199
Epoch 11 | Loss: 0.3975
Epoch 12 | Loss: 0.3763
Epoch 13 | Loss: 0.3554
Epoch 14 | Loss: 0.3349
```

When the training is finished - model can be persisted on S3 bucket.

```

Name                               Last Modified
virus_simulation_output             7 minutes ago
model.pb                           seconds ago

Model save



[50]: torch.save(net.state_dict(), '../resources/model.pb')

[ ]:

[50]: torch.save(net.state_dict(), '../resources/model.pb')

[57]: import boto3
     s3 = boto3.client('s3')

     s3.upload_file("../resources/model.pb", "sagemaker-studio-jho9wmem64", "output/model.pb")
```

Overview			
Type a prefix and press Enter to search. Press ESC to clear.			
<div>Upload Create folder Download Actions</div>			
US East (N. Virg)			
Viewing			
<input type="checkbox"/> Name	Last modified	Size	Storage class
<input type="checkbox"/> 	--	--	--
<input type="checkbox"/>  model.pb	Jun 29, 2020 7:23:49 PM GMT+0200	3.1 KB	Standard
Viewing			