AI with Azure Machine Learning services: Simplifying the data science process

Tony Lin
Mahesh Yadav
agenda

- building your first machine learning model
- working well with others
- showing your work while taking advantage of cloud resources
- optimizing your model
- choosing the right model
- (BONUS TRACK) deploying your model
Azure AI

AI apps & agents
- Azure Bot Service
- Azure Cognitive Services

Machine learning
- Azure Databricks
- Azure Machine Learning

Knowledge mining
- Azure Cognitive Search
## Machine Learning on Azure

<table>
<thead>
<tr>
<th>Feature</th>
<th>Azure Services</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Sophisticated pretrained models</strong></td>
<td>Vision, Speech, Language, Azure Search</td>
</tr>
<tr>
<td>To simplify solution development</td>
<td></td>
</tr>
<tr>
<td><strong>Popular frameworks</strong></td>
<td>Pytorch, TensorFlow, Keras, Onnx</td>
</tr>
<tr>
<td>To build advanced deep learning solutions</td>
<td></td>
</tr>
<tr>
<td><strong>Productive services</strong></td>
<td>Azure Databricks, Azure Machine Learning, Machine Learning VMs</td>
</tr>
<tr>
<td>To empower data science and development teams</td>
<td></td>
</tr>
<tr>
<td><strong>Powerful infrastructure</strong></td>
<td>CPU, GPU, FPGA</td>
</tr>
<tr>
<td>To accelerate deep learning</td>
<td></td>
</tr>
<tr>
<td><strong>Flexible deployment</strong></td>
<td>On-premises, Cloud, Edge</td>
</tr>
<tr>
<td>To deploy and manage models on intelligent cloud and edge</td>
<td></td>
</tr>
</tbody>
</table>
Your first ML model
Making a machine learning model
programming
machine learning

answer

input

algorithm
machine learning

data \rightarrow \text{science-y things} \rightarrow \text{model}
science

question
  ↓
research
  ↓
hypothesis
  ↓
test
  ↓
analyze
  ↓
report
science

question

research

hypothesis

test

analyze

report
working with others
the team workspace - logical

Workspace

- compute
- experiments
- data stores
- models
- images
- deployment
the team workspace - physical

Workspace

.storage

.key vault

.container registry

.application insights
workspace, data stores, and compute
the team workspace

Workspace

compute

experiments

data stores

models

images

deployment
showing your work
running experiments
the team workspace

Workspace

- compute
- experiments
- data stores

models
images
deployment
optimizing your model
hyperdrive in action
choosing the right model
automatic machine learning
(BONUS TRACK) deploying your model
# register model from the best run
model = best_run.register_model(model_name = 'mnist_tf_model', model_path = 'outputs/model')

Note the registered model automatically gets an auto-increasing version number.

```python
print(model.name, model.id, model.version, sep = ' t')
mnist_tf_model  mnist_tf_model:2  2
```

Download the model folder locally and inspect the files in it.

```python
import os
model.download('./')
os.listdir('./model/')
```

```
['mnist-tf.model.meta',
 'checkpoint',
 'mnist-tf.model.index',
 'mnist-tf.model.data-00000-of-00001']
```
register model

mnist_tf_model

Attributes

Version 1
ID mnist_tf_model:1
Date Registered 09/16/2018, 5:22:29 PM UTC
Location aml://asset/7e525521b9614308a993...
Description
Tags
Create images

Models
  +
  Scoring file.py
  +
  Python Environment

→

Images
```python
import json
import numpy as np
import os
import tensorflow as tf

# from azureml.assets.persistence.persistence import get_model_path
from azureml.core.model import Model

def init():
    global X, output, sess
    tf.reset_default_graph()
    # retrieve the local path to the model using the model name
    model_root = Model.get_model_path('mnist_tf_model')
saver = tf.train.import_meta_graph(os.path.join(model_root, 'mnist-tf.model.meta'))
X = tf.get_default_graph().get_tensor_by_name("network/X:0")
output = tf.get_default_graph().get_tensor_by_name("network/output/MatMul:0")
    sess = tf.Session()
    saver.restore(sess, os.path.join(model_root, 'mnist-tf.model'))

def run(raw_data):
    data = np.array(json.loads(raw_data)['data'])
    # make prediction
    out = output.eval(session=sess, feed_dict={X: data})
y_hat = np.argmax(out, axis=1)
    return json.dumps(y_hat.tolist())
```
Environment File

# Conda environment specification. The dependencies defined in this file will
# be automatically provisioned for runs with userManagedDependencies=False.

# Details about the Conda environment file format:
# https://conda.io/docs/user-guide/tasks/manage-environments.html#create-env-file-manually

name: project_environment
dependencies:
  # The python interpreter version.
  # Currently Azure ML only supports 3.5.2 and later.
  - python=3.6.2

  - pip:
    # Required packages for AzureML execution, history, and data preparation.
    - azureml-defaults
    - tensorflow==1.9.0
Create the Image

```python
from azureml.core.image import Image, ContainerImage

image_config = ContainerImage.image_configuration(runtime="python",
    execution_script="score.py",
    conda_file="myenv.yml")

image = Image.create(name="mnist_image",
    models=[model],
    image_config=image_config,
    workspace=ws) # this is the model object
```
Images

The image displays a UI interface with a focus on the 'tf-mnist' project. The interface includes various tabs such as 'Experiments', 'Compute', 'Models', 'Images', 'Deployments', and 'Activities'.

Under the 'Images' tab, the UI shows a list of attributes for the 'tf-mnist' image:

- **Description**: Not specified
- **ID**: tf-mnist:1
- **Date Registered**: 09/16/2018, 5:24:27 PM UTC
- **Version**: 1
- **Location**: danielsacrancaesv.azuredcr.io/tf-mnist
- **Environment**
  - **Type**: Docker
  - **Status**: Succeeded
Deploy image

models

+ Scoring file.py
+ Python Environment

images

+ Azure Kubernetes Service (AKS)
  Or
+ Azure Container Instance

Deployment
from azureml.core.webservice import Webservice
from azureml.core.webservice import AciWebservice

aciconfig = AciWebservice.deploy_configuration(cpu_cores = 1,
                                            memory_gb = 2)

aci_service = Webservice.deploy_from_image(deployment_config = aciconfig,
                                            image = image,
                                            name = 'aci_service_name',
                                            workspace = ws)
Deploy image

ACI deployment of MNIST Model

State: Healthy

Compute Type: ACI

Service ID: aci-service-mnist

Tags: mnist

Creation date: 09/16/2018, 5:52:18 PM UTC

Last updated: 09/16/2018, 5:52:27 PM UTC

Image ID: tf-mnist:1

Scoring URI: http://40.121.221.168:80/score

CPU: 0.1

Memory: 0.5 GB
Deploy image

models

+ Scoring file.py
+ Python Environment

images

+ Azure Kubernetes Service (AKS)
  Or
+ Azure Container Instance

≥ deployment
review

• building your first machine learning model
  • downloaded standard model

• working well with others
  • workspaces, data stores, compute, experiments

• showing your work while taking advantage of cloud resources
  • data stores, compute, and experiments in action (with logging)

• optimizing your model
  • hyperdrive

• choosing the right model
  • automatic machine learning

• (BONUS TRACK) deploying your model
  • model management and deployment
the team workspace

Workspace

- compute
- experiments
- data stores
- models
- images
- deployment
Try it for free

Learn more: http://aka.ms/azureml-docs
Start now: https://azure.microsoft.com/en-us/free/

questions?