ECG_classification

November 20, 2022

```
[ ]: pwd
```

```
[ ]: '/content'
```

```
[ ]: !pip install keras
     !pip install tensorflow
```

```
Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-
wheels/public/simple/
Requirement already satisfied: keras in /usr/local/lib/python3.7/dist-packages
(2.9.0)
Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-
wheels/public/simple/
Requirement already satisfied: tensorflow in /usr/local/lib/python3.7/dist-
packages (2.9.2)
Requirement already satisfied: opt-einsum>=2.3.2 in
/usr/local/lib/python3.7/dist-packages (from tensorflow) (3.3.0)
Requirement already satisfied: keras<2.10.0,>=2.9.0rc0 in
/usr/local/lib/python3.7/dist-packages (from tensorflow) (2.9.0)
Requirement already satisfied: tensorflow-io-gcs-filesystem>=0.23.1 in
/usr/local/lib/python3.7/dist-packages (from tensorflow) (0.27.0)
Requirement already satisfied: keras-preprocessing>=1.1.1 in
/usr/local/lib/python3.7/dist-packages (from tensorflow) (1.1.2)
Requirement already satisfied: setuptools in /usr/local/lib/python3.7/dist-
packages (from tensorflow) (57.4.0)
Requirement already satisfied: libclang>=13.0.0 in
/usr/local/lib/python3.7/dist-packages (from tensorflow) (14.0.6)
Requirement already satisfied: packaging in /usr/local/lib/python3.7/dist-
packages (from tensorflow) (21.3)
Requirement already satisfied: typing-extensions>=3.6.6 in
/usr/local/lib/python3.7/dist-packages (from tensorflow) (4.1.1)
Requirement already satisfied: protobuf<3.20,>=3.9.2 in
/usr/local/lib/python3.7/dist-packages (from tensorflow) (3.19.6)
Requirement already satisfied: google-pasta>=0.1.1 in
/usr/local/lib/python3.7/dist-packages (from tensorflow) (0.2.0)
Requirement already satisfied: astunparse>=1.6.0 in
/usr/local/lib/python3.7/dist-packages (from tensorflow) (1.6.3)
Requirement already satisfied: absl-py>=1.0.0 in /usr/local/lib/python3.7/dist-
```

```
packages (from tensorflow) (1.3.0)
Requirement already satisfied: h5py>=2.9.0 in /usr/local/lib/python3.7/dist-
packages (from tensorflow) (3.1.0)
Requirement already satisfied: grpcio<2.0,>=1.24.3 in
/usr/local/lib/python3.7/dist-packages (from tensorflow) (1.50.0)
Requirement already satisfied: tensorboard<2.10,>=2.9 in
/usr/local/lib/python3.7/dist-packages (from tensorflow) (2.9.1)
Requirement already satisfied: gast<=0.4.0,>=0.2.1 in
/usr/local/lib/python3.7/dist-packages (from tensorflow) (0.4.0)
Requirement already satisfied: wrapt>=1.11.0 in /usr/local/lib/python3.7/dist-
packages (from tensorflow) (1.14.1)
Requirement already satisfied: flatbuffers<2,>=1.12 in
/usr/local/lib/python3.7/dist-packages (from tensorflow) (1.12)
Requirement already satisfied: six>=1.12.0 in /usr/local/lib/python3.7/dist-
packages (from tensorflow) (1.15.0)
Requirement already satisfied: numpy>=1.20 in /usr/local/lib/python3.7/dist-
packages (from tensorflow) (1.21.6)
Requirement already satisfied: termcolor>=1.1.0 in
/usr/local/lib/python3.7/dist-packages (from tensorflow) (2.1.0)
Requirement already satisfied: tensorflow-estimator<2.10.0,>=2.9.0rc0 in
/usr/local/lib/python3.7/dist-packages (from tensorflow) (2.9.0)
Requirement already satisfied: wheel<1.0,>=0.23.0 in
/usr/local/lib/python3.7/dist-packages (from astunparse>=1.6.0->tensorflow)
(0.38.3)
Requirement already satisfied: cached-property in /usr/local/lib/python3.7/dist-
packages (from h5py>=2.9.0->tensorflow) (1.5.2)
Requirement already satisfied: requests<3,>=2.21.0 in
/usr/local/lib/python3.7/dist-packages (from tensorboard<2.10,>=2.9->tensorflow)
(2.23.0)
Requirement already satisfied: tensorboard-data-server<0.7.0,>=0.6.0 in
/usr/local/lib/python3.7/dist-packages (from tensorboard<2.10,>=2.9->tensorflow)
(0.6.1)
Requirement already satisfied: tensorboard-plugin-wit>=1.6.0 in
/usr/local/lib/python3.7/dist-packages (from tensorboard<2.10,>=2.9->tensorflow)
(1.8.1)
Requirement already satisfied: werkzeug>=1.0.1 in /usr/local/lib/python3.7/dist-
packages (from tensorboard<2.10,>=2.9->tensorflow) (1.0.1)
Requirement already satisfied: google-auth<3,>=1.6.3 in
/usr/local/lib/python3.7/dist-packages (from tensorboard<2.10,>=2.9->tensorflow)
(2.14.1)
Requirement already satisfied: markdown>=2.6.8 in /usr/local/lib/python3.7/dist-
packages (from tensorboard<2.10,>=2.9->tensorflow) (3.4.1)
Requirement already satisfied: google-auth-oauthlib<0.5,>=0.4.1 in
/usr/local/lib/python3.7/dist-packages (from tensorboard<2.10,>=2.9->tensorflow)
(0.4.6)
Requirement already satisfied: pyasn1-modules>=0.2.1 in
/usr/local/lib/python3.7/dist-packages (from google-
auth<3,>=1.6.3->tensorboard<2.10,>=2.9->tensorflow) (0.2.8)
```

```
Requirement already satisfied: rsa<5,>=3.1.4 in /usr/local/lib/python3.7/dist-
packages (from google-auth<3,>=1.6.3->tensorboard<2.10,>=2.9->tensorflow) (4.9)
Requirement already satisfied: cachetools<6.0,>=2.0.0 in
/usr/local/lib/python3.7/dist-packages (from google-
auth<3,>=1.6.3->tensorboard<2.10,>=2.9->tensorflow) (5.2.0)
Requirement already satisfied: requests-oauthlib>=0.7.0 in
/usr/local/lib/python3.7/dist-packages (from google-auth-
oauthlib<0.5,>=0.4.1->tensorboard<2.10,>=2.9->tensorflow) (1.3.1)
Requirement already satisfied: importlib-metadata>=4.4 in
/usr/local/lib/python3.7/dist-packages (from
markdown>=2.6.8->tensorboard<2.10,>=2.9->tensorflow) (4.13.0)
Requirement already satisfied: zipp>=0.5 in /usr/local/lib/python3.7/dist-
packages (from importlib-
metadata>=4.4->markdown>=2.6.8->tensorboard<2.10,>=2.9->tensorflow) (3.10.0)
Requirement already satisfied: pyasn1<0.5.0,>=0.4.6 in
/usr/local/lib/python3.7/dist-packages (from pyasn1-modules>=0.2.1->google-
auth<3,>=1.6.3->tensorboard<2.10,>=2.9->tensorflow) (0.4.8)
Requirement already satisfied: certifi>=2017.4.17 in
/usr/local/lib/python3.7/dist-packages (from
requests<3,>=2.21.0->tensorboard<2.10,>=2.9->tensorflow) (2022.9.24)
Requirement already satisfied: chardet<4,>=3.0.2 in
/usr/local/lib/python3.7/dist-packages (from
requests<3,>=2.21.0->tensorboard<2.10,>=2.9->tensorflow) (3.0.4)
Requirement already satisfied: urllib3!=1.25.0,!=1.25.1,<1.26,>=1.21.1 in
/usr/local/lib/python3.7/dist-packages (from
requests<3,>=2.21.0->tensorboard<2.10,>=2.9->tensorflow) (1.24.3)
Requirement already satisfied: idna<3,>=2.5 in /usr/local/lib/python3.7/dist-
packages (from requests<3,>=2.21.0->tensorboard<2.10,>=2.9->tensorflow) (2.10)
Requirement already satisfied: oauthlib>=3.0.0 in /usr/local/lib/python3.7/dist-
packages (from requests-oauthlib>=0.7.0->google-auth-
oauthlib<0.5,>=0.4.1->tensorboard<2.10,>=2.9->tensorflow) (3.2.2)
Requirement already satisfied: pyparsing!=3.0.5,>=2.0.2 in
/usr/local/lib/python3.7/dist-packages (from packaging->tensorflow) (3.0.9)
```

[ ]:

```python
import keras
keras.__version__
```

[ ]:

```
'2.9.0'
```

[ ]:

```python
import tensorflow
tensorflow.__version__
```

[ ]:

```
'2.9.2'
```

[ ]:

```python
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense
from tensorflow.keras.layers import Convolution2D
from tensorflow.keras.layers import MaxPooling2D
from tensorflow.keras.layers import Flatten
```

[ ]:
```
from tensorflow.keras.preprocessing.image import ImageDataGenerator
```

[ ]:
```
train_datagen = ImageDataGenerator(rescale = 1./255,shear_range = 0.
 ↪2,zoom_range = 0.2,horizontal_flip = True)
test_datagen = ImageDataGenerator(rescale = 1./255)
```

[ ]:
```
import os, types
import pandas as pd
from botocore.client import Config
import ibm_boto3

def __iter__(self): return 0

# The following code accesses a file in your IBM Cloud Object Storage. It␣
 ↪includes your credentials.
# You might want to remove those credentials before you share the notebook.
cos_client = ibm_boto3.client(service_name='s3',
    ibm_api_key_id='8VA5EhF5GMnF-ngEHtRQK2IDilQdI0ufj0r54D5pCCZA',
    ibm_auth_endpoint="https://iam.cloud.ibm.com/oidc/token",
    config=Config(signature_version='oauth'),
    endpoint_url='https://s3.private.us.cloud-object-storage.appdomain.cloud')

bucket = 'ecgmodel-donotdelete-pr-nplmu2albo80rd'
object_key = 'data.zip'

streaming_body_1 = cos_client.get_object(Bucket=bucket, Key=object_key)['Body']

# Your data file was loaded into a botocore.response.StreamingBody object.
# Please read the documentation of ibm_boto3 and pandas to learn more about the␣
 ↪possibilities to load the data.
# ibm_boto3 documentation: https://ibm.github.io/ibm-cos-sdk-python/
# pandas documentation: http://pandas.pydata.org/
```

[ ]:
```
from io import BytesIO
import zipfile
unzip=zipfile.ZipFile(BytesIO(streaming_body_1.read()),'r')
file_paths=unzip.namelist()
for path in file_paths:
    unzip.extract(path)
```

[ ]:
```
x_train = train_datagen.flow_from_directory(directory=r'/content/data/
 ↪train',target_size = (64,64),batch_size = 32,class_mode = "categorical")
x_test = test_datagen.flow_from_directory(directory=r'/content/data/
 ↪test',target_size = (64,64),batch_size = 32,class_mode = "categorical")
```

```
Found 15341 images belonging to 6 classes.
Found 6825 images belonging to 6 classes.
```

[ ]:
```
x_train.class_indices
```

```
{'Left Bundle Branch Block': 0,
'Normal': 1,
'Premature Atrial Contraction': 2,
'Premature Ventricular Contractions': 3,
'Right Bundle Branch Block': 4,
'Ventricular Fibrillation': 5}
```

[ ]:
```
model = Sequential()
```

[ ]:
```
model.add(Convolution2D(32,(3,3),input_shape = (64,64,3),activation = "relu"))
```

[ ]:
```
model.add(MaxPooling2D(pool_size = (2,2)))
```

[ ]:
```
model.add(Convolution2D(32,(3,3),activation='relu'))
```

[ ]:
```
model.add(MaxPooling2D(pool_size=(2,2)))
```

[ ]:
```
model.add(Flatten()) # ANN Input...
```

[ ]:
```
model.add(Dense(units = 128,kernel_initializer = "random_uniform",activation =␣
 ↪"relu"))
```

[ ]:
```
model.add(Dense(units = 128,kernel_initializer = "random_uniform",activation =␣
 ↪"relu"))
```

[ ]:
```
model.add(Dense(units = 128,kernel_initializer = "random_uniform",activation =␣
 ↪"relu"))
```

[ ]:
```
model.add(Dense(units = 128,kernel_initializer = "random_uniform",activation =␣
 ↪"relu"))
```

[ ]:
```
model.add(Dense(units = 128,kernel_initializer = "random_uniform",activation =␣
 ↪"relu"))
```

[ ]:
```
model.add(Dense(units = 6,kernel_initializer = "random_uniform",activation =␣
 ↪"softmax"))
```

[ ]:
```
model.summary()
```

```
Model: "sequential_1" ________________________________________________________________________________
Layer (type) Output Shape Param #
=================================================================================
conv2d_2 (Conv2D) (None, 62, 62, 32) 896

max_pooling2d_2 (MaxPooling (None, 31, 31, 32) 0
2D)

conv2d_3 (Conv2D) (None, 29, 29, 32) 9248

max_pooling2d_3 (MaxPooling (None, 14, 14, 32) 0
2D)

flatten_1 (Flatten) (None, 6272) 0

dense_6 (Dense) (None, 128) 802944

dense_7 (Dense) (None, 128) 16512

dense_8 (Dense) (None, 128) 16512

dense_9 (Dense) (None, 128) 16512

dense_10 (Dense) (None, 128) 16512

dense_11 (Dense) (None, 6) 774

=================================================================================
Total params: 879,910 Trainable params: 879,910 Non-trainable params: 0
```

```
[ ]: model.
      ↪compile(optimizer='adam',loss='categorical_crossentropy',metrics=['accuracy'])
```

```
[ ]: model.fit_generator(generator=x_train,steps_per_epoch = len(x_train), epochs=9,␣
      ↪validation_data=x_test,validation_steps = len(x_test))
```

```
Epoch 1/9 480/480 [==============================] - 40s 81ms/step
- loss: 1.4434 - accuracy: 0.4779 - val_loss: 1.6792 - val_accuracy: 0.3193
Epoch 2/9 480/480 [==============================] - 40s 82ms/step
- loss: 1.1352 - accuracy: 0.5648 - val_loss: 1.2390 - val_accuracy: 0.4917
Epoch 3/9 480/480 [==============================] - 39s 81ms/step
- loss: 0.5816 - accuracy: 0.7831 - val_loss: 0.9456 - val_accuracy: 0.7199
Epoch 4/9 480/480 [==============================] - 38s 80ms/step
- loss: 0.2888 - accuracy: 0.9006 - val_loss: 0.7329 - val_accuracy: 0.7979
Epoch 5/9 480/480 [==============================] - 38s 79ms/step
- loss: 0.1966 - accuracy: 0.9378 - val_loss: 0.4952 - val_accuracy: 0.8538
Epoch 6/9 480/480 [==============================] - 38s 80ms/step -
loss: 0.1479 - accuracy: 0.9540 - val_loss: 0.5871 - val_accuracy: 0.8441 Epoch
7/9 480/480 [==============================] - 39s 82ms/step - loss:
0.1267 - accuracy: 0.9609 - val_loss: 0.4398 - val_accuracy: 0.8557 Epoch 8/9
480/480 [==============================] - 39s 82ms/step - loss: 0.1062
- accuracy: 0.9662 - val_loss: 0.6554 - val_accuracy: 0.8593 Epoch 9/9 480/480
```

```
[==============================] - 38s 80ms/step - loss: 0.1003 - accuracy:
0.9685 - val_loss: 0.5559 - val_accuracy: 0.8759
```

```
[ ]: #Saving Model.
     model.save('ECG.h5')
```

```
[ ]: !tar -zcvf ECG-model_new.tgz ECG.h5
```

```
ECG.h5
```

```
[ ]: ls -1
```

```
data/
ECG
ECG-arrhythmia-classification-model_new.tgz
ECG.h5
ECG-model_new.tgz
ECG-model.tgz
```

```
[ ]: pip install watson-machine-learning-client --upgrade
```

```
[ ]: # Replace the credentials that you got from watson machine learning service
     from ibm_watson_machine_learning import APIClient
     wml_credentials = {
                          "url": "https://us-south.ml.cloud.ibm.com",
                          "apikey":"T2vtAoHG5tyBqLPq-gE11hROKW307DoQN1m5SfSHdD9M"
                        }
     client = APIClient(wml_credentials)
```

```
[ ]: client = APIClient(wml_credentials)
```

```
[ ]: client.spaces.list()
```

```
Note: 'limit' is not provided. Only first 50 records will be displayed if the number of records exceed 50
ID                                    NAME                     CREATED
a5359809-7795-48ec-aa02-5bdd98bc7c2b  ECG_imageclassification  2022-11-14T17:43:44.414Z
aadd9634-6c96-4253-b644-512338430797  ibm_deploy               2022-11-05T17:19:15.321Z
```

```
[ ]: def guid_from_space_name(client, space_name):
         space = client.spaces.get_details()
         #print(space)
         return(next(item for item in space['resources'] if item['entity']['name']␣
      ↪== space_name)['metadata']['id'])
```

[ ]:
```
space_uid = guid_from_space_name(client, 'ECG_imageclassification')
print("Space UID = "+ space_uid)
```

```
Space UID = a5359809-7795-48ec-aa02-5bdd98bc7c2b
```

[ ]:
```
client.set.default_space(space_uid)
```

```
’SUCCESS’
```

[ ]:
```
client.software_specifications.list(limit=100)
```

[ ]:
```
software_spec_uid = client.software_specifications.
 ↪get_uid_by_name("tensorflow_rt22.1-py3.9")
software_spec_uid
```

```
’acd9c798-6974-5d2f-a657-ce06e986df4d’
```

[ ]:
```
model_details = client.repository.
 ↪store_model(model='ECG-arrhythmia-classification-model_new.tgz',meta_props={
    client.repository.ModelMetaNames.NAME:"ECG_Model",
    client.repository.ModelMetaNames.TYPE:"tensorflow_2.7",
    client.repository.ModelMetaNames.SOFTWARE_SPEC_UID:software_spec_uid})
model_id=client.repository.get_model_uid(model_details)
```

```
This method is deprecated, please use get_model_id()
```

[ ]:
```
model_id
```

```
’23c336b9-b8c4-425a-85d8-808b78fd3b95’
```

[ ]:
```
client.repository.download(model_id,'my_modell.tar1.gz')
```

```
Successfully saved model content to file: ’my_modell.tar1.gz’
```

```
’/home/wsuser/work/my_modell.tar1.gz’
```

[ ]:
```
from tensorflow.keras.models import load_model
from tensorflow.keras.preprocessing import image
```

[ ]:
```
model = load_model("ECG.h5")
```