ECG_Classification_model

November 20, 2022

ECG arrhythmia classification using CNN

[ ]:
```
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: grpcio<2.0,>=1.24.3 in
/usr/local/lib/python3.7/dist-packages (from tensorflow) (1.50.0)
Requirement already satisfied: astunparse>=1.6.0 in
/usr/local/lib/python3.7/dist-packages (from tensorflow) (1.6.3)
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: numpy>=1.20 in /usr/local/lib/python3.7/dist-
packages (from tensorflow) (1.21.6)
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: typing-extensions>=3.6.6 in
/usr/local/lib/python3.7/dist-packages (from tensorflow) (4.1.1)
Requirement already satisfied: google-pasta>=0.1.1 in
/usr/local/lib/python3.7/dist-packages (from tensorflow) (0.2.0)
Requirement already satisfied: absl-py>=1.0.0 in /usr/local/lib/python3.7/dist-
packages (from tensorflow) (1.3.0)
Requirement already satisfied: termcolor>=1.1.0 in
```

```
/usr/local/lib/python3.7/dist-packages (from tensorflow) (2.1.0)
Requirement already satisfied: six>=1.12.0 in /usr/local/lib/python3.7/dist-
packages (from tensorflow) (1.15.0)
Requirement already satisfied: keras-preprocessing>=1.1.1 in
/usr/local/lib/python3.7/dist-packages (from tensorflow) (1.1.2)
Requirement already satisfied: h5py>=2.9.0 in /usr/local/lib/python3.7/dist-
packages (from tensorflow) (3.1.0)
Requirement already satisfied: setuptools in /usr/local/lib/python3.7/dist-
packages (from tensorflow) (57.4.0)
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: libclang>=13.0.0 in
/usr/local/lib/python3.7/dist-packages (from tensorflow) (14.0.6)
Requirement already satisfied: flatbuffers<2,>=1.12 in
/usr/local/lib/python3.7/dist-packages (from tensorflow) (1.12)
Requirement already satisfied: tensorboard<2.10,>=2.9 in
/usr/local/lib/python3.7/dist-packages (from tensorflow) (2.9.1)
Requirement already satisfied: wrapt>=1.11.0 in /usr/local/lib/python3.7/dist-
packages (from tensorflow) (1.14.1)
Requirement already satisfied: packaging in /usr/local/lib/python3.7/dist-
packages (from tensorflow) (21.3)
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: 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: 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: 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: 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: 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: 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: 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: 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: 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: 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: 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: 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)
```

[ ]:
```
import keras
keras.__version__
```

[ ]:
```
'2.9.0'
```

[ ]:
```
import tensorflow
tensorflow.__version__
```

[ ]:
```
'2.9.2'
```

Importing Neccessary Libraries

[ ]:
```
import numpy as np#used for numerical analysis
import tensorflow #open source used for both ML and DL for computation
from tensorflow.keras.models import Sequential #it is a plain stack of layers
from tensorflow.keras import layers #A layer consists of a tensor-in tensor-out␣
↪computation function
#Dense layer is the regular deeply connected neural network layer
from tensorflow.keras.layers import Dense,Flatten
#Faltten-used fot flattening the input or change the dimension
from tensorflow.keras.layers import Conv2D,MaxPooling2D #Convolutional layer
#MaxPooling2D-for downsampling the image
from keras.preprocessing.image import ImageDataGenerator
```

Image Data Agumentation

[ ]:
```
#setting parameter for Image Data agumentation to the traing data
train_datagen=ImageDataGenerator(rescale=1./255,shear_range=0.2,zoom_range=0.
↪2,horizontal_flip=True)
```

[ ]:
```
#Image Data agumentation to the testing data
test_datagen=ImageDataGenerator(rescale=1./255)
```

Loading our data and performing data agumentation

[ ]:
```
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.

if os.environ.get('RUNTIME_ENV_LOCATION_TYPE') == 'external':
    endpoint_69c05974e5c84795a978662af2736fc1 = 'https://s3.us.
↪cloud-object-storage.appdomain.cloud'
else:
    endpoint_69c05974e5c84795a978662af2736fc1 = 'https://s3.private.us.
↪cloud-object-storage.appdomain.cloud'

client_69c05974e5c84795a978662af2736fc1 = ibm_boto3.client(service_name='s3',
    ibm_api_key_id='',
    ibm_auth_endpoint="https://iam.cloud.ibm.com/oidc/token",
    config=Config(signature_version='oauth'),
    endpoint_url=endpoint_69c05974e5c84795a978662af2736fc1)
```

```
streaming_body_1 = client_69c05974e5c84795a978662af2736fc1.
 ↪get_object(Bucket='ecgimagebasedheartbeatclassificat-donotdelete-pr-l2ugdciyflqayf',␣
 ↪Key='data.zip')['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()
print(file_paths)
for path in file_paths:
    unzip.extract(path)
```

[ ]:
```
pwd
```

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

[ ]:
```
import os
filenames = os.listdir('/content')
```

[ ]:
```
#performing data agumentation to train data
x_train=train_datagen.flow_from_directory('/content/data/
 ↪train',target_size=(64,64),batch_size=32,class_mode='categorical')
#performing data agumentation to test data
x_test=test_datagen.flow_from_directory('/conent/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.

[ ]:
```
print(x_train.class_indices)#checking the number of classes
```

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

[ ]:
```
from collections import Counter as c
c(x_train.labels)
```

Counter({0: 504, 1: 7346, 2: 2054, 3: 2759, 4: 2239, 5: 439})

Creating the model

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

```
# adding model layer
model.
 ↪add(Conv2D(32,(3,3),input_shape=(64,64,3),activation='relu'))#convolutional␣
 ↪layer
model.add(MaxPooling2D(pool_size=(2,2))) #MaxPooling2D-for downsampling the␣
 ↪input

model.add(Conv2D(32,(3,3),activation='relu'))
model.add(MaxPooling2D(pool_size=(2,2)))

model.add(Flatten())#flatten the dimension of the image
model.add(Dense(32))#deeply connected neural network layers.
model.add(Dense(6,activation='softmax'))#output layer with 6 neurons
```

[ ]:
```
model.summary()#summary of our model
```

Model: ”sequential” _____
Layer (type) Output Shape Param #
==========
conv2d (Conv2D) (None, 62, 62, 32) 896

max_pooling2d (MaxPooling2D (None, 31, 31, 32) 0
)

conv2d_1 (Conv2D) (None, 29, 29, 32) 9248

max_pooling2d_1 (MaxPooling (None, 14, 14, 32) 0
2D)

flatten (Flatten) (None, 6272) 0

dense (Dense) (None, 32) 200736

dense_1 (Dense) (None, 6) 198

==========
Total params: 211,078 Trainable params: 211,078 Non-trainable params: 0

Compiling the model

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

Fitting the model

[ ]:
```
# Fit the model
model.fit_generator(generator=x_train,steps_per_epoch = len(x_train),
                    epochs=10, validation_data=x_test,validation_steps =␣
 ↪len(x_test))
```

Epoch 1/10 480/480 [==========] - 97s 201ms/step - loss: 0.7450 - accuracy: 0.7491 - val_loss: 0.4413 - val_accuracy: 0.8504 Epoch 2/10 480/480 [==========] - 96s 201ms/step - loss: 0.2724 - accuracy: 0.9186 - val_loss: 0.4429 - val_accuracy: 0.8809 Epoch 3/10 480/480 [==========] - 97s 201ms/step - loss: 0.2204 - accuracy: 0.9347 - val_loss: 0.2756 - val_accuracy: 0.9207 Epoch 4/10 480/480 [==========] - 97s 201ms/step - loss: 0.1896 - accuracy: 0.9435 - val_loss: 0.2367 - val_accuracy: 0.9314 Epoch 5/10 480/480 [==========] - 96s 201ms/step - loss: 0.1637 - accuracy: 0.9497 - val_loss: 0.2607 - val_accuracy: 0.9174 Epoch 6/10 480/480 [==========] - 96s 200ms/step - loss: 0.1526 - accuracy: 0.9544 - val_loss: 0.2423 - val_accuracy: 0.9263 Epoch 7/10 480/480 [==========] - 96s 199ms/step - loss: 0.1366 - accuracy: 0.9567 - val_loss: 0.2960 - val_accuracy: 0.9169 Epoch 8/10 480/480 [==========] - 97s 202ms/step - loss: 0.1232 - accuracy: 0.9608 - val_loss: 0.3316 - val_accuracy: 0.9078 Epoch 9/10 480/480 [==========] - 97s 201ms/step - loss: 0.1255 - accuracy: 0.9626 - val_loss: 0.2454 - val_accuracy: 0.9297 Epoch 10/10 480/480 [==========] - 95s 199ms/step - loss: 0.1174 - accuracy: 0.9638 - val_loss: 0.3265 - val_accuracy: 0.9182

[ ]:
```
# model.fit_generator(x_train,epochs=10,validation_data=x_test)
```

Saving our model

[ ]:
```
# Save the model
from tensorflow.keras.models import load_model
model.save('ECG.h5')
```

[ ]:
```
!tar -zcvf ECG-Image-based-heartbeat-classification-model_new.tgz ECG.h5
```

ECG.h5

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

data/

ECG.h5

ECG-Image-based-heartbeat-classification-model_new.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":"WpWfHtY_VXAXAgD4uzxBAb00C2FJstDEVyb3oXk1UaRm"
                  }
```

```
client = APIClient(wml_credentials)
```

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

[ ]:
```
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, 'image_classification')
print("Space UID = "+ space_uid)
```

Space UID = 26b6a24d-f745-4d09-b456-8f6dfd7d9ca6

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

'SUCCESS'

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

NAME ASSET_ID TYPE default_py3.6 0062b8c9-8b7d-44a0-a9b9-46c416adcbd9 base kernel-spark3.2-scala2.12 020d69ce-7ac1-5e68-ac1a-31189867356a base pytorch-onnx_1.3-py3.7-edt 069ea134-3346-5748-b513-49120e15d288 base scikit-learn_0.20-py3.6 09c5a1d0-9c1e-4473-a344-eb7b665ff687 base spark-mllib_3.0-scala_2.12 09f4cff0-90a7-5899-b9ed-1ef348aebdee base pytorch-onnx_rt22.1-py3.9 0b848dd4-e681-5599-be41-b5f6fccc6471 base ai-function_0.1-py3.6 0cdb0f1e-5376-4f4d-92dd-da3b69aa9bda base shiny-r3.6 0e6e79df-875e-4f24-8ae9-62dcc2148306 base tensorflow_2.4-py3.7-horovod 1092590a-307d-563d-9b62-4eb7d64b3f22 base pytorch_1.1-py3.6 10ac12d6-6b30-4ccd-8392-3e922c096a92 base tensorflow_1.15-py3.6-ddl 111e41b3-de2d-5422-a4d6-bf776828c4b7 base runtime-22.1-py3.9 12b83a17-24d8-5082-900f-0ab31fbfd3cb base scikit-learn_0.22-py3.6 154010fa-5b3b-4ac1-82af-4d5ee5abbc85 base default_r3.6 1b70aec3-ab34-4b87-8aa0-a4a3c8296a36 base pytorch-onnx_1.3-py3.6 1bc6029a-cc97-56da-b8e0-39c3880dbbe7 base pytorch-onnx_rt22.1-py3.9-edt 1d362186-7ad5-5b59-8b6c-9d0880bde37f base tensorflow_2.1-py3.6 1eb25b84-d6ed-5dde-b6a5-3fbdf1665666 base spark-mllib_3.2 20047f72-0a98-58c7-9ff5-a77b012eb8f5 base tensorflow_2.4-py3.8-horovod 217c16f6-178f-56bf-824a-b19f20564c49 base runtime-22.1-py3.9-cuda 26215f05-08c3-5a41-a1b0-da66306ce658 base do_py3.8 295addb5-9ef9-547e-9bf4-92ae3563e720 base autoai-ts_3.8-py3.8 2aa0c932-798f-5ae9-abd6-15e0c2402fb5 base tensorflow_1.15-py3.6 2b73a275-7cbf-420b-a912-eae7f436e0bc base pytorch_1.2-py3.6 2c8ef57d-2687-4b7d-acce-01f94976dac1 base spark-mllib_2.3 2e51f700-bca0-4b0d-88dc-5c6791338875 base pytorch-onnx_1.1-py3.6-edt 32983cea-3f32-4400-8965-dde874a8d67e base spark-mllib_3.0-py37 36507ebe-8770-55ba-ab2a-eafe787600e9 base spark-mllib_2.4 390d21f8-e58b-4fac-9c55-d7ceda621326 base xgboost_0.82-py3.6 39e31acd-5f30-41dc-ae44-60233c80306e base pytorch-onnx_1.2-py3.6-edt 40589d0e-7019-4e28-8daa-fb03b6f4fe12 base default_r36py38 41c247d3-45f8-5a71-b065-8580229facf0 base autoai-ts_rt22.1-py3.9 4269d26e-07ba-5d40-8f66-2d495b0c71f7 base autoai-obm_3.0 42b92e18-d9ab-567f-988a-4240ba1ed5f7 base pmml-3.0_4.3 493bcb95-16f1-5bc5-bee8-81b8af80e9c7 base spark-mllib_2.4-r_3.6 49403dff-92e9-4c87-a3d7-a42d0021c095 base xgboost_0.90-py3.6 4ff8d6c2-1343-4c18-85e1-689c965304d3 base pytorch-onnx_1.1-py3.6 50f95b2a-bc16-43bb-bc94-b0bed208c60b base

[ ]:
```
import tensorflow
tensorflow.__version__
```

[ ]:
```
'2.9.2'
```

[ ]:
```
software_spec_uid = client.software_specifications.
 ↪get_uid_by_name("tensorflow_2.4-py3.7")
software_spec_uid
```

'65e171d7-72d1-55d9-8ebb-f813d620c9bb'

[ ]:
```
model_details = client.repository.
 ↪store_model(model='ECG-Image-based-heartbeat-classification-model_new.
 ↪tgz',meta_props={
client.repository.ModelMetaNames.NAME:"image_classification",
client.repository.ModelMetaNames.SOFTWARE_SPEC_UID: client.
 ↪software_specifications.get_uid_by_name("tensorflow_2.4-py3.7"),
client.repository.ModelMetaNames.TYPE:"keras_2.2.5"})

model_id = client.repository.get_model_uid(model_details)
```

[ ]:
```
client.repository.download(model_id, 'my_model.tar.gz')
```

Successfully saved model content to file: 'my_model.tar.gz'

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

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