Audio files - Deep learning

```
#audio files
!wget 'http://storage.googleapis.com/download.tensorflow.org/data/mini_speech_commands.zip
#unzip
!unzip mini_speech_commands.zip
#delete unnecessary files
!rm '/content/mini_speech_commands/README.md'
```

```
  inflating: mini_speech_commands/yes/28ed6bc9_nohash_1.wav
  inflating: __MACOSX/mini_speech_commands/yes/._28ed6bc9_nohash_1.wav
  inflating: mini_speech_commands/yes/e805a617_nohash_0.wav
  inflating: __MACOSX/mini_speech_commands/yes/._e805a617_nohash_0.wav
  inflating: mini_speech_commands/yes/d197e3ae_nohash_3.wav
  inflating: __MACOSX/mini_speech_commands/yes/._d197e3ae_nohash_3.wav
  inflating: mini_speech_commands/yes/bd2db1a5_nohash_0.wav
  inflating: __MACOSX/mini_speech_commands/yes/._bd2db1a5_nohash_0.wav
  inflating: mini_speech_commands/yes/50f55535_nohash_1.wav
  inflating: __MACOSX/mini_speech_commands/yes/._50f55535_nohash_1.wav
  inflating: mini_speech_commands/yes/f550b7dc_nohash_0.wav
  inflating: __MACOSX/mini_speech_commands/yes/._f550b7dc_nohash_0.wav
  inflating: mini_speech_commands/yes/1aeef15e_nohash_1.wav
  inflating: __MACOSX/mini_speech_commands/yes/._1aeef15e_nohash_1.wav
  inflating: mini_speech_commands/yes/a0f93943_nohash_1.wav
  inflating: __MACOSX/mini_speech_commands/yes/._a0f93943_nohash_1.wav
  inflating: mini_speech_commands/yes/ab7b5acd_nohash_0.wav
  inflating: __MACOSX/mini_speech_commands/yes/._ab7b5acd_nohash_0.wav
  inflating: mini_speech_commands/yes/baeac2ba_nohash_3.wav
  inflating: __MACOSX/mini_speech_commands/yes/._baeac2ba_nohash_3.wav
  inflating: mini_speech_commands/yes/28ce0c58_nohash_3.wav
  inflating: __MACOSX/mini_speech_commands/yes/._28ce0c58_nohash_3.wav
  inflating: mini_speech_commands/yes/617de221_nohash_2.wav
  inflating: __MACOSX/mini_speech_commands/yes/._617de221_nohash_2.wav
  inflating: mini_speech_commands/yes/d0faf7e4_nohash_0.wav
  inflating: __MACOSX/mini_speech_commands/yes/._d0faf7e4_nohash_0.wav
  inflating: mini_speech_commands/yes/e649aa92_nohash_0.wav
  inflating: __MACOSX/mini_speech_commands/yes/._e649aa92_nohash_0.wav
  inflating: mini_speech_commands/yes/e7ea8b76_nohash_0.wav
  inflating: __MACOSX/mini_speech_commands/yes/._e7ea8b76_nohash_0.wav
  inflating: mini_speech_commands/yes/459345ea_nohash_0.wav
  inflating: __MACOSX/mini_speech_commands/yes/._459345ea_nohash_0.wav
  inflating: mini_speech_commands/yes/b97c9f77_nohash_3.wav
  inflating: __MACOSX/mini_speech_commands/yes/._b97c9f77_nohash_3.wav
  inflating: mini_speech_commands/yes/ec201020_nohash_0.wav
  inflating: __MACOSX/mini_speech_commands/yes/._ec201020_nohash_0.wav
  inflating: mini_speech_commands/yes/24c9f572_nohash_2.wav
  inflating: __MACOSX/mini_speech_commands/yes/._24c9f572_nohash_2.wav
  inflating: mini_speech_commands/yes/7d8babdb_nohash_0.wav
  inflating: __MACOSX/mini_speech_commands/yes/._7d8babdb_nohash_0.wav
  inflating: mini_speech_commands/yes/3006c271_nohash_0.wav
  inflating: __MACOSX/mini_speech_commands/yes/._3006c271_nohash_0.wav
  inflating: mini_speech_commands/yes/7799c9cd_nohash_0.wav
  inflating: __MACOSX/mini_speech_commands/yes/._7799c9cd_nohash_0.wav
  inflating: mini_speech_commands/yes/b7a0754f_nohash_1.wav
  inflating: __MACOSX/mini_speech_commands/yes/._b7a0754f_nohash_1.wav
  inflating: mini_speech_commands/yes/ad63d93c_nohash_0.wav
  inflating: __MACOSX/mini_speech_commands/yes/._ad63d93c_nohash_0.wav
  inflating: mini_speech_commands/yes/c2aeb59d_nohash_0.wav
  inflating: __MACOSX/mini_speech_commands/yes/._c2aeb59d_nohash_0.wav
  inflating: mini_speech_commands/yes/7cbf645a_nohash_0.wav
  inflating: __MACOSX/mini_speech_commands/yes/._7cbf645a_nohash_0.wav
  inflating: mini_speech_commands/yes/30802c5d_nohash_0.wav
  inflating: __MACOSX/mini_speech_commands/yes/._30802c5d_nohash_0.wav
  inflating: mini_speech_commands/yes/da2c5f1b_nohash_2.wav
  inflating: __MACOSX/mini_speech_commands/yes/._da2c5f1b_nohash_2.wav
  inflating: mini_speech_commands/yes/c0c0d87d_nohash_0.wav
  inflating: __MACOSX/mini_speech_commands/yes/._c0c0d87d_nohash_0.wav
```

```
#pip install librosa
```

```
import os
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import librosa
from scipy.io import wavfile
import IPython.display as ipd
```

```
ipd.Audio('/content/mini_speech_commands/down/004ae714_nohash_0.wav')
```

0:00 / 0:01

```
ipd.Audio('/content/mini_speech_commands/no/0132a06d_nohash_1.wav')
```

0:00 / 0:01

```
samples, samplingrate = librosa.load('/content/mini_speech_commands/down/004ae714_nohash_
```

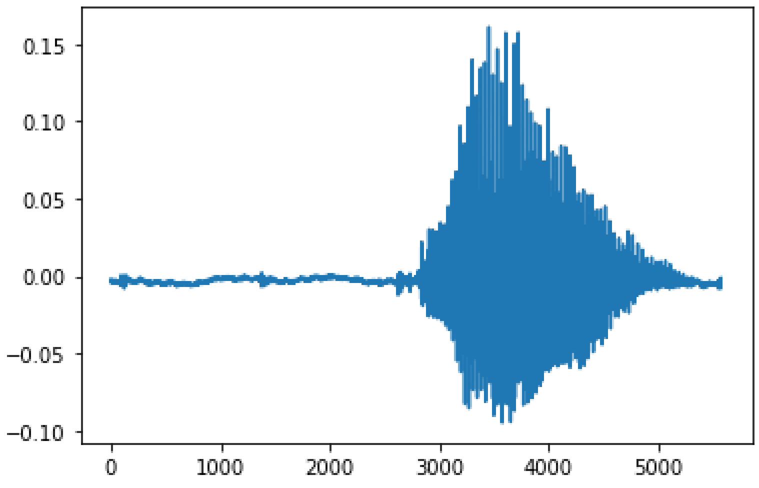
```
plt.plot(samples)
```

```
[<matplotlib.lines.Line2D at 0x7f9f923f45e0>]
```

```
s,sr=librosa.load('/content/mini_speech_commands/go/004ae714_nohash_0.wav', sr=8000)
```

```
plt.plot(s)
```

```
[<matplotlib.lines.Line2D at 0x7f9f9251e0a0>]
```



read audio

```
#samplingrate=X intervals/sec
#samplingrate=8000
#sound length=1sec
#samples=8000

#sound length=2secs
#samples=16000
```

```
X = []
y = []
```

```
def collectAudio(foldername):
  basefolder='/content/mini_speech_commands/'
  folderpath=os.path.join(basefolder,foldername)
  for audio in os.listdir(folderpath):
    filepath = os.path.join(folderpath,audio)
    samples, _ = librosa.load(filepath, sr=8000)
    if(len(samples)==8000):
      X.append(samples)
      y.append(foldername)
```

```
collectAudio('down')
```

```
collectAudio('go')
collectAudio('left')
```

```
collectAudio('no')
```

```
collectAudio('right')
```

```
collectAudio('stop')
```

```
collectAudio('up')
```

```
collectAudio('yes')
```

Feature extraction

```
import librosa

def feature_chromagram(waveform, sample_rate):
    # STFT computed here explicitly; mel spectrogram and MFCC functions do this under the
    stft_spectrogram=np.abs(librosa.stft(waveform))
    # Produce the chromagram for all STFT frames and get the mean of each column of the re
    chromagram=np.mean(librosa.feature.chroma_stft(S=stft_spectrogram, sr=sample_rate).T,a
    return chromagram

def feature_melspectrogram(waveform, sample_rate):
    # Produce the mel spectrogram for all STFT frames and get the mean of each column of t
    # Using 8khz as upper frequency bound should be enough for most speech classification
    melspectrogram=np.mean(librosa.feature.melspectrogram(y=waveform, sr=sample_rate, n_me
    return melspectrogram

def feature_mfcc(waveform, sample_rate):
    # Compute the MFCCs for all STFT frames and get the mean of each column of the resulti
    # 40 filterbanks = 40 coefficients
    mfc_coefficients=np.mean(librosa.feature.mfcc(y=waveform, sr=sample_rate, n_mfcc=40).T
    return mfc_coefficients


def get_features(waveform):
    # load an individual soundfile
        chromagram = feature_chromagram(waveform, sample_rate=8000)
        melspectrogram = feature_melspectrogram(waveform, sample_rate=8000)
        mfc_coefficients = feature_mfcc(waveform, sample_rate=8000)

        feature_matrix=np.array([])
        # use np.hstack to stack our feature arrays horizontally to create a feature matri
        feature_matrix = np.hstack((chromagram, melspectrogram, mfc_coefficients))

        return feature_matrix
```

```
X_feats = []


for audio in X:
  features = get_features(audio)
  X_feats.append(features)
```

```
/usr/local/lib/python3.8/dist-packages/librosa/filters.py:238: UserWarning: Empty fi
  warnings.warn(
/usr/local/lib/python3.8/dist-packages/librosa/core/pitch.py:153: UserWarning: Tryin
  warnings.warn("Trying to estimate tuning from empty frequency set.")
```

```
len(X_feats)
```

```
7178
```

```
len(y)
```

```
7178
```

```
X[0].shape
```

```
(8000,)
```

```
X_np = np.array(X_feats)
```

```
X_np.shape
```

```
(7178, 180)
```

```
#audiocommands = ['down','go','left','no','right','stop','up','yes']
#for i in audiocommands:
#  collectAudio(i)
```

```
#audiocommands = os.listdir('mini_speech_commands')
#for i in audiocommands:
#  collectAudio(i)
```

```
len(X_feats)
```

```
7178
```

```
X_np = np.array(X_feats)
```

```
from sklearn.preprocessing import LabelEncoder
```

```
from tensorflow.keras.utils import to_categorical
```

```
enc = LabelEncoder()
enc.fit(y)
y_le = enc.transform(y)
y_one = to_categorical(y_le)
```

```
enc.classes_
```

```
array(['down', 'go', 'left', 'no', 'right', 'stop', 'up', 'yes'],
      dtype='<U5')
```

```
X_np.shape
```

```
(7178, 180)
```

```
y_one
```

```
array([[1., 0., 0., ..., 0., 0., 0.],
       [1., 0., 0., ..., 0., 0., 0.],
       [1., 0., 0., ..., 0., 0., 0.],
       ...,
       [0., 0., 0., ..., 0., 0., 1.],
       [0., 0., 0., ..., 0., 0., 1.],
       [0., 0., 0., ..., 0., 0., 1.]], dtype=float32)
```

```
from sklearn.preprocessing import StandardScaler
sc = StandardScaler()
```

```
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X_np, y_one, test_size=0.2, shuffle=Tr
```

```
X_train, X_val, y_train, y_val = train_test_split(X_train, y_train, test_size=0.25, strati
```

```
sc.fit(X_train)
```

```
StandardScaler()
```

```
sc.transform(X_train)
sc.transform(X_val)
sc.transform(X_test)
```

```
array([[ 1.0109284 ,  0.97579086,  0.8512493 , ...,  0.4357058 ,
        -0.49280354,  0.46771193],
       [ 1.0139474 ,  0.48520976,  0.2896098 , ...,  0.43405348,
         0.6980211 , -0.28553087],
       [ 0.07905947, -0.70210993, -0.69571346, ...,  1.3153495 ,
         2.662557  ,  2.0867739 ],
       ...,
       [-1.8213546 , -1.6026284 , -2.275829  , ..., -0.15214053,
         0.12827682,  0.19066845],
       [-0.2260332 , -0.32264674, -0.00390457, ...,  1.2267808 ,
         0.2556669 ,  0.95220494],
       [ 0.8823554 ,  0.18664744,  0.5132361 , ...,  0.38027427,
         1.1937288 ,  1.3058097 ]], dtype=float32)
```

```
X_train.shape
```

```
(4306, 180)
```

```
X_val.shape
```

```
(1436, 180)
```

```
X_test.shape
```

```
(1436, 180)
```

ANN

```
from keras.models import Sequential, load_model
from keras.layers import Dense, Dropout
```

```
audioANN = Sequential()
```

```
audioANN.add(Dense(units=512, activation='relu',input_dim=180))
audioANN.add(Dropout(rate=0.25))
audioANN.add(Dense(units=8, activation='softmax'))
```

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

```
from keras.callbacks import EarlyStopping, ModelCheckpoint, ReduceLROnPlateau

es = EarlyStopping(monitor='val_accuracy', min_delta=0, patience=30, verbose=1, mode='autc
mc = ModelCheckpoint(filepath='bestweights.h5', monitor='val_accuracy', verbose=1, save_be
rd = ReduceLROnPlateau(monitor='val_accuracy', factor=0.1, patience=15, verbose=1, mode='a
```

```
history = audioANN.fit(X_train, y_train, validation_data=(X_val, y_val), epochs=100, callb
```

```
135/135 [==============================] - 0s 3ms/step - loss: 0.3728 - accuracy:
Epoch 87/100
118/135 [========================>....] - ETA: 0s - loss: 0.3864 - accuracy: 0.8
Epoch 87: val_accuracy did not improve from 0.51184
135/135 [==============================] - 0s 3ms/step - loss: 0.3838 - accuracy:
Epoch 88/100
133/135 [============================>.] - ETA: 0s - loss: 0.3778 - accuracy: 0.8
Epoch 88: val_accuracy did not improve from 0.51184
135/135 [==============================] - 1s 4ms/step - loss: 0.3775 - accuracy:
Epoch 89/100
```

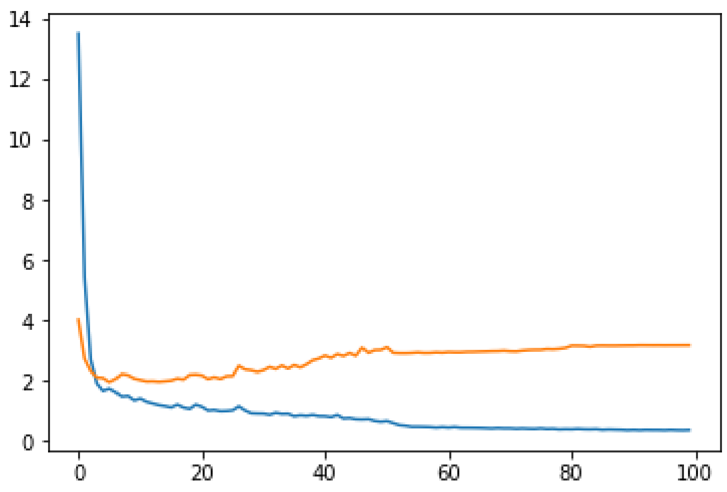
```
121/135 [=========================>....] - ETA: 0s - loss: 0.3675 - accuracy: 0.8
Epoch 89: val_accuracy did not improve from 0.51184
135/135 [==============================] - 0s 3ms/step - loss: 0.3736 - accuracy:
Epoch 90/100
119/135 [=========================>....] - ETA: 0s - loss: 0.3585 - accuracy: 0.8
Epoch 90: val_accuracy did not improve from 0.51184
135/135 [==============================] - 0s 3ms/step - loss: 0.3612 - accuracy:
Epoch 91/100
133/135 [============================>.] - ETA: 0s - loss: 0.3686 - accuracy: 0.8
Epoch 91: val_accuracy did not improve from 0.51184
135/135 [==============================] - 1s 4ms/step - loss: 0.3706 - accuracy:
Epoch 92/100
135/135 [==============================] - ETA: 0s - loss: 0.3607 - accuracy: 0.8
Epoch 92: val_accuracy did not improve from 0.51184
135/135 [==============================] - 0s 4ms/step - loss: 0.3607 - accuracy:
Epoch 93/100
119/135 [=========================>....] - ETA: 0s - loss: 0.3745 - accuracy: 0.8
Epoch 93: val_accuracy did not improve from 0.51184
135/135 [==============================] - 1s 4ms/step - loss: 0.3714 - accuracy:
Epoch 94/100
133/135 [============================>.] - ETA: 0s - loss: 0.3661 - accuracy: 0.8
Epoch 94: val_accuracy did not improve from 0.51184
135/135 [==============================] - 1s 4ms/step - loss: 0.3663 - accuracy:
Epoch 95/100
120/135 [=========================>....] - ETA: 0s - loss: 0.3652 - accuracy: 0.8
Epoch 95: val_accuracy did not improve from 0.51184
135/135 [==============================] - 1s 4ms/step - loss: 0.3665 - accuracy:
Epoch 96/100
131/135 [============================>.] - ETA: 0s - loss: 0.3543 - accuracy: 0.8
Epoch 96: val_accuracy did not improve from 0.51184
135/135 [==============================] - 1s 4ms/step - loss: 0.3593 - accuracy:
Epoch 97/100
129/135 [===========================>..] - ETA: 0s - loss: 0.3704 - accuracy: 0.8
Epoch 97: val_accuracy did not improve from 0.51184
135/135 [==============================] - 0s 4ms/step - loss: 0.3693 - accuracy:
Epoch 98/100
132/135 [============================>.] - ETA: 0s - loss: 0.3609 - accuracy: 0.8
Epoch 98: val_accuracy did not improve from 0.51184
135/135 [==============================] - 0s 3ms/step - loss: 0.3619 - accuracy:
Epoch 99/100
117/135 [=========================>....] - ETA: 0s - loss: 0.3542 - accuracy: 0.8
Epoch 99: val_accuracy did not improve from 0.51184
135/135 [==============================] - 1s 4ms/step - loss: 0.3572 - accuracy:
Epoch 100/100
133/135 [============================>.] - ETA: 0s - loss: 0.3625 - accuracy: 0.8
Epoch 100: val_accuracy did not improve from 0.51184
135/135 [==============================] - 1s 4ms/step - loss: 0.3641 - accuracy:
```

```
newmodel = load_model('bestweights.h5')
```

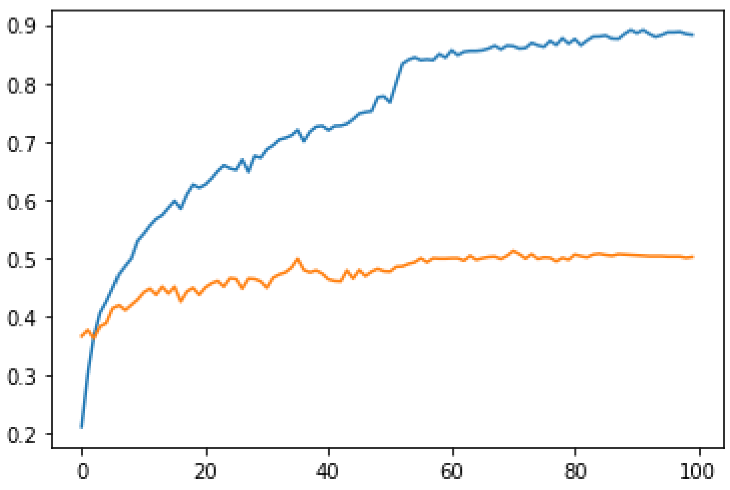
```
newmodel.evaluate(X_test, y_test)
```

```
45/45 [==============================] - 0s 2ms/step - loss: 3.0367 - accuracy: 0.50
[3.0367062091827393, 0.5062674283981323]
```

```
plt.plot(history.history['loss'])
plt.plot(history.history['val_loss'])
plt.show()
```



```
plt.plot(history.history['accuracy'])
plt.plot(history.history['val_accuracy'])
plt.show()
```



Predicting on Xtest

```
ypred = audioANN.predict(X_test)
```

```
45/45 [==============================] - 0s 1ms/step
```

```
ypred
```

```
array([[5.0489124e-10, 1.0917451e-07, 4.6525670e-06, ..., 2.2241954e-08,
        4.8933278e-11, 9.9999523e-01],
       [5.4835004e-04, 6.9849318e-01, 4.9186889e-03, ..., 1.8403770e-01,
        1.5751328e-02, 5.4666668e-04],
       [7.7206796e-01, 1.6543398e-08, 6.8160295e-03, ..., 2.0056289e-02,
        1.9935597e-01, 1.2850921e-09],
       ...,
       [3.5506392e-01, 1.8414650e-04, 3.1661619e-02, ..., 2.1840412e-09,
        2.8489927e-10, 1.7132412e-01],
       [2.3233704e-02, 9.2007779e-03, 2.7653411e-01, ..., 9.2149625e-04,
        5.5739924e-04, 6.7494768e-01],
       [6.8805757e-04, 1.1953798e-01, 1.8653271e-03, ..., 4.6171360e-02,
        8.0584556e-01, 1.8373326e-05]], dtype=float32)
```

```
import numpy as np
ypredclasses = np.argmax(ypred, axis=-1)
```

```
ypredclasses
```

```
array([7, 1, 0, ..., 0, 7, 6])
```

```
y_actual = enc.inverse_transform(ypredclasses)
```

```
y_actual
```

```
array(['yes', 'go', 'down', ..., 'down', 'yes', 'up'], dtype='<U5')
```

```
audioANN.evaluate(X_test, y_test)
```

```
45/45 [==============================] - 0s 2ms/step - loss: 3.2196 - accuracy: 0.51
[3.2195613384246826, 0.5139275789260864]
```

```
newmodel.evaluate(X_test, y_test)
```

```
45/45 [==============================] - 0s 2ms/step - loss: 3.0367 - accuracy: 0.50
[3.0367062091827393, 0.5062674283981323]
```

Deployment - website

```
!pip install flask gevent requests pillow flask-ngrok pyngrok
```

```
Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-wheels/
Requirement already satisfied: flask in /usr/local/lib/python3.8/dist-packages (1.1.
Collecting gevent
  Downloading gevent-22.10.2-cp38-cp38-manylinux_2_17_x86_64.manylinux2014_x86_64.wh
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 6.5/6.5 MB 81.6 MB/s eta 0:00:00
Requirement already satisfied: requests in /usr/local/lib/python3.8/dist-packages (2
Requirement already satisfied: pillow in /usr/local/lib/python3.8/dist-packages (7.1
Collecting flask-ngrok
  Downloading flask_ngrok-0.0.25-py3-none-any.whl (3.1 kB)
Collecting pyngrok
```

```
  Downloading pyngrok-5.2.1.tar.gz (761 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 761.3/761.3 KB 57.8 MB/s eta 0:00:00
  Preparing metadata (setup.py) ... done
Requirement already satisfied: Jinja2<3.0,>=2.10.1 in /usr/local/lib/python3.8/dist-
Requirement already satisfied: click<8.0,>=5.1 in /usr/local/lib/python3.8/dist-pack
Requirement already satisfied: Werkzeug<2.0,>=0.15 in /usr/local/lib/python3.8/dist-
Requirement already satisfied: itsdangerous<2.0,>=0.24 in /usr/local/lib/python3.8/d
Collecting zope.interface
  Downloading zope.interface-5.5.2-cp38-cp38-manylinux_2_5_x86_64.manylinux1_x86_64.
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 261.4/261.4 KB 29.4 MB/s eta 0:00:00
Collecting zope.event
  Downloading zope.event-4.6-py2.py3-none-any.whl (6.8 kB)
Requirement already satisfied: greenlet>=2.0.0 in /usr/local/lib/python3.8/dist-pack
Requirement already satisfied: setuptools in /usr/local/lib/python3.8/dist-packages
Requirement already satisfied: idna<3,>=2.5 in /usr/local/lib/python3.8/dist-package
Requirement already satisfied: chardet<5,>=3.0.2 in /usr/local/lib/python3.8/dist-pa
Requirement already satisfied: urllib3<1.27,>=1.21.1 in /usr/local/lib/python3.8/dis
Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.8/dist-p
Requirement already satisfied: PyYAML in /usr/local/lib/python3.8/dist-packages (fro
Requirement already satisfied: MarkupSafe>=0.23 in /usr/local/lib/python3.8/dist-pac
Building wheels for collected packages: pyngrok
  Building wheel for pyngrok (setup.py) ... done
  Created wheel for pyngrok: filename=pyngrok-5.2.1-py3-none-any.whl size=19792 sha2
  Stored in directory: /root/.cache/pip/wheels/5d/f2/70/526da675d32f17577ec47ac4c663
Successfully built pyngrok
Installing collected packages: zope.interface, zope.event, pyngrok, gevent, flask-ng
Successfully installed flask-ngrok-0.0.25 gevent-22.10.2 pyngrok-5.2.1 zope.event-4.
```

```python
from flask_ngrok import run_with_ngrok
from flask import Flask, render_template, request
from tensorflow.keras.preprocessing.image import load_img, img_to_array
import pyngrok
from pyngrok import ngrok
```

```python
procfile = 'web: gunicorn app:app'
procfiles = open('/content/Procfile', 'w')
```

```python
procfiles.write(procfile)
```

```
21
```

```python
procfiles.close()
```

```python
!mkdir '/content/templates'
```

```python
!mkdir '/content/uploads'
```

```
'''
index.html
```

```
<!doctype html>
<html lang="en">
<head>
 <title> Recognition Server</title>
  <body>

      <form  action="" method="post" enctype=multipart/form-data>
        <input type="text" name="textbox" placeholder="Movie review" required="required" /
        <input type=submit value=Upload>
    </form>

    <h3>Prediction is</h3>
    {{label}}


  </body>
</html>


'''
```

```
'\nindex.html\n\n\n<!doctype html>\n<html lang="en">\n<head>\n <title> Recognition
Server</title>\n  <body>\n      \n      <form  action="" method="post" enctype=mult
ipart/form-data>\n       \t<input type="text" name="textbox" placeholder="Movie rev
iew" required="required" />\n        <input type=submit value=Upload>\n    </form>
```

Connecting webpage with ANN

```
import pyngrok
```

```
import os
```

```
'''
ngrok.com - authtoken and below
'''
!ngrok authtoken
```

```
NAME:
   authtoken - save authtoken to configuration file

USAGE:
   ngrok authtoken [command options] [arguments...]

DESCRIPTION:
   The authtoken command modifies your configuration file to include
   the specified authtoken. By default, this configuration file is located
   at $HOME/.ngrok2/ngrok.yml

   The ngrok.com service requires that you sign up for an account to use
   many advanced service features. In order to associate your client with
   an account, it must pass a secret token to the ngrok.com service when it
   starts up. Instead of passing this authtoken on every invocation, you may
   use this command to save it into your configuration file so that your
   client always authenticates you properly.

EXAMPLE:
    ngrok authtoken BDZIXnhJt2HNWLXyQ5PM_qCaBq0W2sNFcCa0rfTZd

OPTIONS:
   --config             save in this config file, default: ~/.ngrok2/ngrok.yml
   --log "false"        path to log file, 'stdout', 'stderr' or 'false'
   --log-format "term"  log record format: 'term', 'logfmt', 'json'
   --log-level "info"   logging level

ERROR:  You must pass a single argument, the authtoken to save to configuration file
```

```
app = Flask(__name__)
run_with_ngrok(app)

app.config['UPLOADS'] = '/content/uploads'

@app.route('/')
def home():
  return render_template('index.html')

@app.route('/', methods=['POST'])
def prediction():textbox
data = request.files['/content/mini_speech_commands/down/004ae714_nohash_0.wav']
filepath = os.path.join(app.config['UPLOADS'],data.filename)
data.save(filepath)


samples, _ = librosa.load(filepath, sr=8000)
features = get_features(samples)
features = np.reshape(features,(1,180,1))
preds = bestmodel.predict(features)
label = np.argmax(preds, axis=1)

classes = enc.inverse_transform(label)

return render_template('index.html', label=classes)

if __name__=='__main__':
    app.run()
```

```
---------------------------------------------------------------------------
RuntimeError                              Traceback (most recent call last)
<ipython-input-87-12f118999145> in <module>
     10 @app.route('/', methods=['POST'])
     11 def prediction():textbox
---> 12 data =
request.files['/content/mini_speech_commands/down/004ae714_nohash_0.wav']
     13 filepath = os.path.join(app.config['UPLOADS'],data.filename)
     14 data.save(filepath)
```

2 frames

```
/usr/local/lib/python3.8/dist-packages/flask/globals.py in _lookup_req_object(name)


---> 38         raise RuntimeError(_request_ctx_err_msg)
     39     return getattr(top, name)
     40

RuntimeError: Working outside of request context.

This typically means that you attempted to use functionality that needed
an active HTTP request.  Consult the documentation on testing for
information about how to avoid this problem.
```

SEARCH STACK OVERFLOW

Colab paid products - Cancel contracts here

0s completed at 5:07 PM