MFCC 10 sec Model Creation

May 22, 2023

[1]:
```
import os
import json
import csv
import sys
import numpy as np
import matplotlib.pyplot as plt
import librosa
import tensorflow as tf
from sklearn.model_selection import train_test_split, StratifiedShuffleSplit
from tensorflow.keras.utils import plot_model
import datetime
import time
from tensorflow.keras.callbacks import TensorBoard
```

```
2023-05-16 22:04:42.397162: I tensorflow/core/platform/cpu_feature_guard.cc:182]
This TensorFlow binary is optimized to use available CPU instructions in
performance-critical operations.
To enable the following instructions: AVX2 FMA, in other operations, rebuild
TensorFlow with the appropriate compiler flags.
```

[2]:
```
csv_path = "csv_fma_10secs_data.csv"
sr=22050
csv.field_size_limit(sys.maxsize)
```

[2]:
```
131072
```

[3]:
```
#Read MFCC data from CSV
def csv_read_data(csv_path):
    # Load data from CSV file
    with open(csv_path, 'r') as csvfile:
        reader = csv.reader(csvfile)

        # Skip the header row
        next(reader)

        # Initialize lists to hold genre and MFCC data
        genres = []
        mfcc = []
```

```
        # Iterate over each row of the CSV file
        for row in reader:
            # Extract genre and MFCC data from the row
            genre = int(row[0])
            mfcc_data = json.loads(row[1])

            # Append genre and MFCC data to lists
            genres.append(genre)
            mfcc.append(mfcc_data)

    # Convert lists to numpy arrays
    X = np.array(mfcc)
    y = np.array(genres)

    return X, y
```

[4]:
```
def stratified_split_dataset(inputs, targets, test_split_size, val_split_size):
    sss = StratifiedShuffleSplit(n_splits=1, test_size=test_split_size,␣
↪random_state=42)

    train_val_indices, test_indices = next(sss.split(inputs, targets))
    inputs_train_val, targets_train_val = inputs[train_val_indices],␣
↪targets[train_val_indices]

    val_size = val_split_size / (1 - test_split_size)
    sss_train = StratifiedShuffleSplit(n_splits=1, test_size=val_size,␣
↪random_state=43)
    train_indices, val_indices = next(sss_train.split(inputs_train_val,␣
↪targets_train_val))

    inputs_train, targets_train = inputs_train_val[train_indices],␣
↪targets_train_val[train_indices]
    inputs_val, targets_val = inputs_train_val[val_indices],␣
↪targets_train_val[val_indices]
    inputs_test, targets_test = inputs[test_indices], targets[test_indices]

    # Needed for compatibility reasons
    inputs_train = inputs_train[..., np.newaxis]
    inputs_val = inputs_val[..., np.newaxis]
    inputs_test=  inputs_test[..., np.newaxis]

    return inputs_train, inputs_val, inputs_test, targets_train, targets_val,␣
↪targets_test
```

[5]:
```
#Used to plot performance history of model
def plot_performance(hist):

    acc = hist.history['acc']
    val_acc = hist.history['val_acc']
    loss = hist.history['loss']
    val_loss = hist.history['val_loss']

    epochs = range(len(acc))

    plt.plot(epochs, acc, 'r', label='Training accuracy')
    plt.plot(epochs, val_acc, 'b', label='Validation accuracy')
    plt.title('Training and validation accuracy')
    plt.legend()
    plt.figure()

    plt.plot(epochs, loss, 'r', label='Training Loss')
    plt.plot(epochs, val_loss, 'b', label='Validation Loss')
    plt.title('Training and validation loss')
    plt.legend()

    plt.show()
```

[6]:
```
#=============== MAIN ===============
#Reading MFCC data from CSV
inputs, targets = csv_read_data(csv_path)
```

[7]:
```
#Splitting data
Xtrain, Xval, Xtest, ytrain, yval, ytest =stratified_split_dataset(inputs,␣
 ↪targets, 0.15, 0.15)
```

[8]:
```
print(Xtrain.shape)
print(Xtest.shape)
print(Xval.shape)
```

```
(7301, 417, 40, 1)
(1565, 417, 40, 1)
(1565, 417, 40, 1)
```

[9]:
```
#Design the model
def design_model(input_shape):

    model = tf.keras.models.Sequential([

        tf.keras.layers.Conv2D(64, (3,3), activation='relu',␣
 ↪input_shape=input_shape),
        tf.keras.layers.MaxPooling2D((3,3), strides=(2,2), padding='same'),
```

```
        tf.keras.layers.BatchNormalization(),

        tf.keras.layers.Conv2D(64, (3,3), activation='relu'),
        tf.keras.layers.MaxPooling2D((3,3), strides=(2,2), padding='same'),
        tf.keras.layers.BatchNormalization(),

        tf.keras.layers.Conv2D(64, (3,3), activation='relu'),
        tf.keras.layers.MaxPooling2D((3,3), strides=(2,2), padding='same'),
        tf.keras.layers.BatchNormalization(),
        tf.keras.layers.Dropout(0.3),

        tf.keras.layers.Conv2D(64, (3,3), activation='relu'),
        tf.keras.layers.MaxPooling2D((3,3), strides=(2,2), padding='same'),
        tf.keras.layers.Dropout(0.3),

        tf.keras.layers.Flatten(),
        tf.keras.layers.Dense(64, activation='relu'),
        tf.keras.layers.Dense(len(np.unique(targets)), activation='softmax')
    ])

    return model
```

[10]:
```
#Model Creation
input_shape = (Xtrain.shape[1], Xtrain.shape[2], 1)
model = design_model(input_shape)

model.compile(optimizer = tf.keras.optimizers.Adam(learning_rate=0.0001),
              loss = 'sparse_categorical_crossentropy',
              metrics = ['acc'])
```

```
2023-05-16 22:06:34.848421: I
tensorflow/core/common_runtime/gpu/gpu_device.cc:1635] Created device
/job:localhost/replica:0/task:0/device:GPU:0 with 22077 MB memory:  -> device:
0, name: NVIDIA GeForce RTX 4090, pci bus id: 0000:c1:00.0, compute capability:
8.9
```

[11]:
```
#Model Summary
model.summary()
```

```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv2d (Conv2D)             (None, 415, 38, 64)       640

 max_pooling2d (MaxPooling2D  (None, 208, 19, 64)      0
 )
```

```
batch_normalization (BatchN  (None, 208, 19, 64)      256
ormalization)

conv2d_1 (Conv2D)            (None, 206, 17, 64)      36928

max_pooling2d_1 (MaxPooling  (None, 103, 9, 64)       0
2D)

batch_normalization_1 (Batc  (None, 103, 9, 64)       256
hNormalization)

conv2d_2 (Conv2D)            (None, 101, 7, 64)       36928

max_pooling2d_2 (MaxPooling  (None, 51, 4, 64)        0
2D)

batch_normalization_2 (Batc  (None, 51, 4, 64)        256
hNormalization)

dropout (Dropout)            (None, 51, 4, 64)        0

conv2d_3 (Conv2D)            (None, 49, 2, 64)        36928

max_pooling2d_3 (MaxPooling  (None, 25, 1, 64)        0
2D)

dropout_1 (Dropout)          (None, 25, 1, 64)        0

flatten (Flatten)            (None, 1600)             0

dense (Dense)                (None, 64)               102464

dense_1 (Dense)              (None, 6)                390

=================================================================
Total params: 215,046
Trainable params: 214,662
Non-trainable params: 384
_________________________________________________________________
```

[12]:
```
#TENSORBOARD
NAME= "MFCC 10secs Model"
log_dir = "logs/fit/" + NAME
tensorboard_callback = [tf.keras.callbacks.TensorBoard(log_dir=log_dir,␣
 ↪histogram_freq=1),tf.keras.callbacks.EarlyStopping(patience=5,␣
 ↪restore_best_weights=True)]
```

[13]:
```
#COMPILATION
history = model.fit(Xtrain, ytrain,
                    validation_data = (Xval, yval),
                    epochs = 550,
                    batch_size = 64,
                    callbacks=[tensorboard_callback])

model.save("./Models/"+NAME)
```

```
Epoch 1/550
```

```
2023-05-16 22:06:36.657097: E
tensorflow/core/grappler/optimizers/meta_optimizer.cc:954] layout failed:
INVALID_ARGUMENT: Size of values 0 does not match size of permutation 4 @ fanin
shape insequential/dropout/dropout/SelectV2-2-TransposeNHWCToNCHW-
LayoutOptimizer
2023-05-16 22:06:37.598943: I
tensorflow/compiler/xla/stream_executor/cuda/cuda_dnn.cc:424] Loaded cuDNN
version 8600
2023-05-16 22:06:38.333676: I
tensorflow/compiler/xla/stream_executor/cuda/cuda_blas.cc:637] TensorFloat-32
will be used for the matrix multiplication. This will only be logged once.
2023-05-16 22:06:38.505280: I tensorflow/compiler/xla/service/service.cc:169]
XLA service 0x7f2301ddee10 initialized for platform CUDA (this does not
guarantee that XLA will be used). Devices:
2023-05-16 22:06:38.505309: I tensorflow/compiler/xla/service/service.cc:177]
StreamExecutor device (0): NVIDIA GeForce RTX 4090, Compute Capability 8.9
2023-05-16 22:06:38.509082: I
tensorflow/compiler/mlir/tensorflow/utils/dump_mlir_util.cc:269] disabling MLIR
crash reproducer, set env var `MLIR_CRASH_REPRODUCER_DIRECTORY` to enable.
2023-05-16 22:06:38.637358: I ./tensorflow/compiler/jit/device_compiler.h:180]
Compiled cluster using XLA!  This line is logged at most once for the lifetime
of the process.
```

```
115/115 [==============================] - 7s 17ms/step - loss: 1.6261 - acc:
0.4087 - val_loss: 1.4462 - val_acc: 0.4102
Epoch 2/550
115/115 [==============================] - 1s 12ms/step - loss: 1.2304 - acc:
0.5388 - val_loss: 1.1591 - val_acc: 0.5917
Epoch 3/550
115/115 [==============================] - 1s 12ms/step - loss: 1.1372 - acc:
0.5742 - val_loss: 1.0622 - val_acc: 0.6083
Epoch 4/550
115/115 [==============================] - 1s 13ms/step - loss: 1.0800 - acc:
0.5940 - val_loss: 1.0237 - val_acc: 0.6096
Epoch 5/550
115/115 [==============================] - 1s 13ms/step - loss: 1.0405 - acc:
0.6143 - val_loss: 1.0031 - val_acc: 0.6339
```

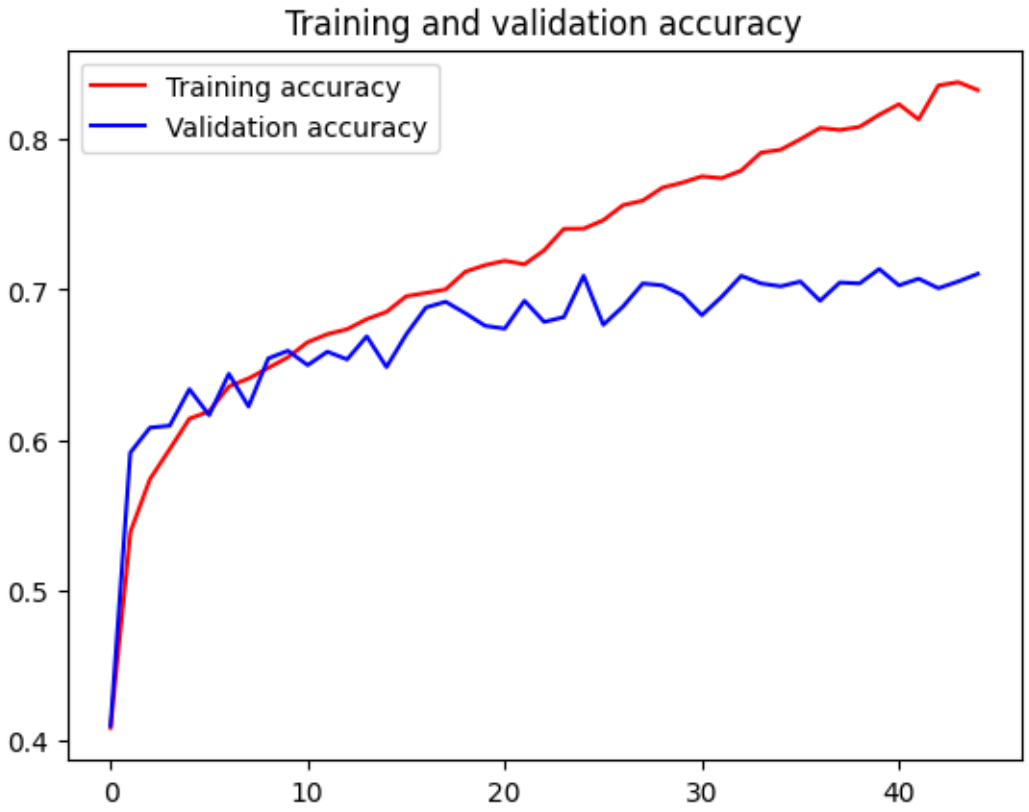
```
Epoch 6/550
115/115 [==============================] - 1s 12ms/step - loss: 1.0134 - acc:
0.6191 - val_loss: 0.9981 - val_acc: 0.6166
Epoch 7/550
115/115 [==============================] - 1s 13ms/step - loss: 0.9813 - acc:
0.6357 - val_loss: 0.9589 - val_acc: 0.6441
Epoch 8/550
115/115 [==============================] - 1s 13ms/step - loss: 0.9629 - acc:
0.6410 - val_loss: 1.0149 - val_acc: 0.6224
Epoch 9/550
115/115 [==============================] - 1s 12ms/step - loss: 0.9514 - acc:
0.6480 - val_loss: 0.9490 - val_acc: 0.6543
Epoch 10/550
115/115 [==============================] - 1s 12ms/step - loss: 0.9259 - acc:
0.6551 - val_loss: 0.9286 - val_acc: 0.6594
Epoch 11/550
115/115 [==============================] - 1s 12ms/step - loss: 0.9031 - acc:
0.6651 - val_loss: 0.9384 - val_acc: 0.6498
Epoch 12/550
115/115 [==============================] - 1s 12ms/step - loss: 0.8888 - acc:
0.6705 - val_loss: 0.9153 - val_acc: 0.6588
Epoch 13/550
115/115 [==============================] - 1s 12ms/step - loss: 0.8655 - acc:
0.6737 - val_loss: 0.9234 - val_acc: 0.6537
Epoch 14/550
115/115 [==============================] - 1s 12ms/step - loss: 0.8500 - acc:
0.6805 - val_loss: 0.9081 - val_acc: 0.6690
Epoch 15/550
115/115 [==============================] - 1s 12ms/step - loss: 0.8484 - acc:
0.6852 - val_loss: 0.9301 - val_acc: 0.6486
Epoch 16/550
115/115 [==============================] - 1s 12ms/step - loss: 0.8148 - acc:
0.6957 - val_loss: 0.9086 - val_acc: 0.6703
Epoch 17/550
115/115 [==============================] - 1s 12ms/step - loss: 0.8062 - acc:
0.6978 - val_loss: 0.8751 - val_acc: 0.6882
Epoch 18/550
115/115 [==============================] - 1s 12ms/step - loss: 0.7860 - acc:
0.7002 - val_loss: 0.8630 - val_acc: 0.6920
Epoch 19/550
115/115 [==============================] - 1s 12ms/step - loss: 0.7743 - acc:
0.7120 - val_loss: 0.8753 - val_acc: 0.6843
Epoch 20/550
115/115 [==============================] - 1s 12ms/step - loss: 0.7559 - acc:
0.7163 - val_loss: 0.8842 - val_acc: 0.6760
Epoch 21/550
115/115 [==============================] - 1s 12ms/step - loss: 0.7530 - acc:
0.7192 - val_loss: 0.8619 - val_acc: 0.6741
```

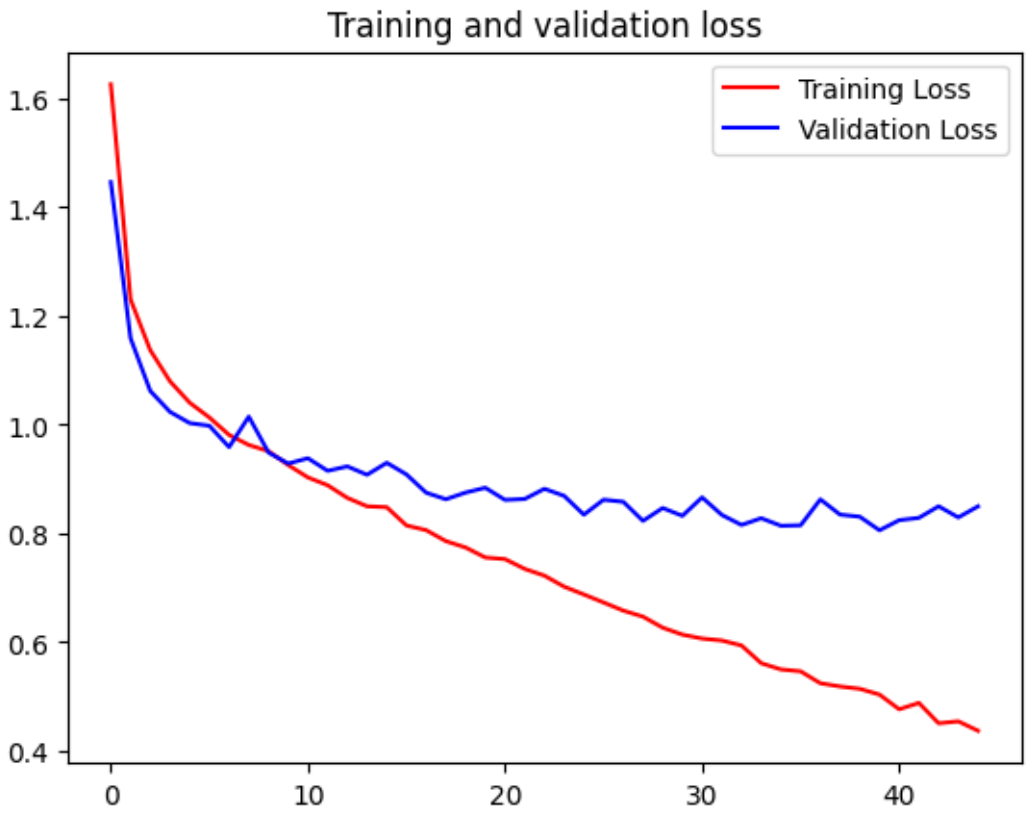
```
Epoch 22/550
115/115 [==============================] - 1s 12ms/step - loss: 0.7350 - acc:
0.7169 - val_loss: 0.8635 - val_acc: 0.6927
Epoch 23/550
115/115 [==============================] - 1s 12ms/step - loss: 0.7225 - acc:
0.7262 - val_loss: 0.8823 - val_acc: 0.6786
Epoch 24/550
115/115 [==============================] - 1s 12ms/step - loss: 0.7023 - acc:
0.7404 - val_loss: 0.8693 - val_acc: 0.6818
Epoch 25/550
115/115 [==============================] - 1s 12ms/step - loss: 0.6879 - acc:
0.7406 - val_loss: 0.8346 - val_acc: 0.7093
Epoch 26/550
115/115 [==============================] - 1s 12ms/step - loss: 0.6732 - acc:
0.7462 - val_loss: 0.8621 - val_acc: 0.6767
Epoch 27/550
115/115 [==============================] - 1s 12ms/step - loss: 0.6583 - acc:
0.7562 - val_loss: 0.8584 - val_acc: 0.6888
Epoch 28/550
115/115 [==============================] - 1s 12ms/step - loss: 0.6472 - acc:
0.7592 - val_loss: 0.8233 - val_acc: 0.7042
Epoch 29/550
115/115 [==============================] - 1s 12ms/step - loss: 0.6269 - acc:
0.7678 - val_loss: 0.8469 - val_acc: 0.7029
Epoch 30/550
115/115 [==============================] - 1s 13ms/step - loss: 0.6141 - acc:
0.7711 - val_loss: 0.8325 - val_acc: 0.6965
Epoch 31/550
115/115 [==============================] - 1s 13ms/step - loss: 0.6069 - acc:
0.7754 - val_loss: 0.8664 - val_acc: 0.6831
Epoch 32/550
115/115 [==============================] - 1s 13ms/step - loss: 0.6033 - acc:
0.7741 - val_loss: 0.8343 - val_acc: 0.6952
Epoch 33/550
115/115 [==============================] - 1s 13ms/step - loss: 0.5942 - acc:
0.7792 - val_loss: 0.8157 - val_acc: 0.7093
Epoch 34/550
115/115 [==============================] - 1s 13ms/step - loss: 0.5615 - acc:
0.7911 - val_loss: 0.8282 - val_acc: 0.7042
Epoch 35/550
115/115 [==============================] - 1s 13ms/step - loss: 0.5501 - acc:
0.7930 - val_loss: 0.8143 - val_acc: 0.7022
Epoch 36/550
115/115 [==============================] - 1s 13ms/step - loss: 0.5466 - acc:
0.8000 - val_loss: 0.8151 - val_acc: 0.7054
Epoch 37/550
115/115 [==============================] - 2s 13ms/step - loss: 0.5247 - acc:
0.8076 - val_loss: 0.8628 - val_acc: 0.6927
```

```
Epoch 38/550
115/115 [==============================] - 1s 13ms/step - loss: 0.5188 - acc:
0.8062 - val_loss: 0.8348 - val_acc: 0.7048
Epoch 39/550
115/115 [==============================] - 1s 12ms/step - loss: 0.5145 - acc:
0.8082 - val_loss: 0.8309 - val_acc: 0.7042
Epoch 40/550
115/115 [==============================] - 1s 12ms/step - loss: 0.5039 - acc:
0.8163 - val_loss: 0.8060 - val_acc: 0.7137
Epoch 41/550
115/115 [==============================] - 1s 12ms/step - loss: 0.4769 - acc:
0.8233 - val_loss: 0.8245 - val_acc: 0.7029
Epoch 42/550
115/115 [==============================] - 1s 13ms/step - loss: 0.4886 - acc:
0.8132 - val_loss: 0.8288 - val_acc: 0.7073
Epoch 43/550
115/115 [==============================] - 1s 13ms/step - loss: 0.4511 - acc:
0.8358 - val_loss: 0.8501 - val_acc: 0.7010
Epoch 44/550
115/115 [==============================] - 1s 13ms/step - loss: 0.4546 - acc:
0.8380 - val_loss: 0.8297 - val_acc: 0.7054
Epoch 45/550
115/115 [==============================] - 1s 12ms/step - loss: 0.4373 - acc:
0.8328 - val_loss: 0.8500 - val_acc: 0.7105

2023-05-16 22:07:45.732154: I tensorflow/core/common_runtime/executor.cc:1197]
[/device:CPU:0] (DEBUG INFO) Executor start aborting (this does not indicate an
error and you can ignore this message): INVALID_ARGUMENT: You must feed a value
for placeholder tensor 'inputs' with dtype float and shape [?,51,4,64]
         [[{{node inputs}}]]
2023-05-16 22:07:45.745121: I tensorflow/core/common_runtime/executor.cc:1197]
[/device:CPU:0] (DEBUG INFO) Executor start aborting (this does not indicate an
error and you can ignore this message): INVALID_ARGUMENT: You must feed a value
for placeholder tensor 'inputs' with dtype float and shape [?,25,1,64]
         [[{{node inputs}}]]
2023-05-16 22:07:46.206991: I tensorflow/core/common_runtime/executor.cc:1197]
[/device:CPU:0] (DEBUG INFO) Executor start aborting (this does not indicate an
error and you can ignore this message): INVALID_ARGUMENT: You must feed a value
for placeholder tensor 'inputs' with dtype float and shape [?,51,4,64]
         [[{{node inputs}}]]
2023-05-16 22:07:46.240386: I tensorflow/core/common_runtime/executor.cc:1197]
[/device:CPU:0] (DEBUG INFO) Executor start aborting (this does not indicate an
error and you can ignore this message): INVALID_ARGUMENT: You must feed a value
for placeholder tensor 'inputs' with dtype float and shape [?,25,1,64]
         [[{{node inputs}}]]
WARNING:absl:Found untraced functions such as _jit_compiled_convolution_op,
_jit_compiled_convolution_op, _jit_compiled_convolution_op,
_jit_compiled_convolution_op while saving (showing 4 of 4). These functions will
```

```
not be directly callable after loading.

INFO:tensorflow:Assets written to: ./Models/MFCC 10secs Model/assets

INFO:tensorflow:Assets written to: ./Models/MFCC 10secs Model/assets
```

[14]:
```
#Plot Performance
plot_performance(history)
```





[15]:
```
y_pred = model.predict(Xtest)
```

```
49/49 [==============================] - 0s 2ms/step
```

[16]:
```
from sklearn.metrics import accuracy_score

y_pred_classes = np.argmax(y_pred, axis=1)

# Calculate the accuracy
accuracy = accuracy_score(ytest, y_pred_classes)
print(f"Test accuracy: {accuracy:.4f}")
```

```
Test accuracy: 0.7016
```

[19]:
```
from sklearn.metrics import f1_score
```

[22]:
```
# Make predictions on the test set
y_pred_test = model.predict(Xtest)

# Convert probabilities to class labels
y_pred_test_labels = np.argmax(y_pred_test, axis=1)
```

```
# Compute the F1 score
f1_test = f1_score(ytest, y_pred_test_labels, average='micro')

print('Test F1 score: ', f1_test)
```

```
49/49 [==============================] - 0s 2ms/step
Test F1 score:  0.7015974440894569
```