The Heart of the Matter

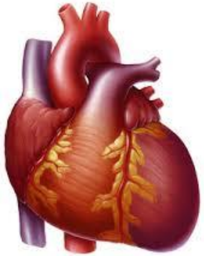
By: Spenser Johnson | Mentor: Koyuki Nakamori

Introduction and dataset

According to the WHO, cardiovascular diseases are the number one cause of death worldwide. Being able to identify some of these disorders would be great for everyone's health. Since the data comes from a digital stethoscope and an iPhone app, this means it is accessible to many people, making it an impactful project.

I found the data set on Kaggle.com, but the original data and challenges come from Peter J. Bentley of the Department of Computer Science at the University College London. It comes in three .csv's and two .zip files containing .wav files. For this project, I'll be identifying different anomalies using deep learning, and building a model that can predict these anomalies.

Classes



- Normal - a clear “lub dub, lub dub” pattern, with the time from “lub” to “dub” shorter than the time from “dub” to the next “lub” (when the heart rate is less than 140 beats per minute)
- Artifact - a wide range of different sounds, including feedback squeals and echoes, speech, music and noise. There are usually no discernable heart sounds
- Murmur - Heart murmurs sound as though there is a “whooshing, roaring, rumbling, or turbulent fluid” noise in one of two temporal locations: (1) between “lub” and “dub”, or (2) between “dub” and “lub”. They can be a symptom of many heart disorders, some serious.
- Extra Heart Sound - can be identified because there is an additional sound, e.g. a “lub-lub dub” or a “lub dub-dub”. An extra heart sound may not be a sign of disease. However, in some situations it is an important sign of disease, which if detected early could help a person.

Preprocessing

- Get_labels - vectorizes .wav files using keras one-hot vectorization to_categorical function
- Wav2mfcc / Wav2chroma - feature extraction from .wav file
- Save_data_to_array - saves features as numpy array in a .npy file
- Get_train_test - train test split
- Prepare / load_dataset

Feature Extraction

- Mel-Frequency Cepstral Coefficients (MFCC)
  - a representation of the short-term power spectrum of a sound
  - based on a linear cosine transform of a log power spectrum on a nonlinear mel scale of frequency
- Chromatic Features
  - the entire spectrum is projected onto 12 bins representing the 12 distinct semitones (or chroma) of the musical octave

Models

- Logistic regression
  - TEST SCORE: 64%
- Support Vector Classifier
  - TEST SCORE: 22%
- Keras Dense
  - TEST SCORE: 54%
- Keras 1DConv
  - TEST SCORE: 57%
- Keras 2DConv with MFCC
  - TEST SCORE: 60%
- Keras 2DConv with Chroma
  - TEST SCORE: 65%

```
def get_model():
    model = Sequential()
    model.add(Conv2D(32, kernel_size=(2, 2), activation='relu', input_shape=(feature_dim_1, feature_dim_2, channel)))
    model.add(Conv2D(48, kernel_size=(2, 2), activation='relu'))
    model.add(Conv2D(120, kernel_size=(2, 2), activation='relu'))
    model.add(MaxPooling2D(pool_size=(2, 2)))
    model.add(Dropout(0.25))
    model.add(Flatten())
    model.add(Dense(128, activation='relu'))
    model.add(Dropout(0.25))
    model.add(Dense(64, activation='relu'))
    model.add(Dropout(0.4))
    model.add(Dense(num_classes, activation='softmax'))
    model.compile(loss=keras.losses.categorical_crossentropy,
                  optimizer=keras.optimizers.Adadelta(),
                  metrics=['accuracy'])
    return model

# Predicts one sample
def predict(filepath, model):
    sample = wav2mfcc(filepath)
    sample_reshaped = sample.reshape(1, feature_dim_1, feature_dim_2, channel)
    return get_labels()[0][
            np.argmax(model.predict(sample_reshaped))
    ]
model.compile(optimizer='rmsprop', loss='binary_crossentropy', metrics=['accuracy'])
```

Future steps

- Build into iPhone app
- Gather more data (only 124 samples)
- Build unsupervised labeling model for new inputs
- Grid search hyperparameters (feature length, learning rate, etc.)
- Explore different audio feature extraction methods (Linear predictive analysis, Power spectral analysis, etc.)

-