Classification of Acoustic Physiological Signals Based on Deep Learning Neural Networks with Augmented Features

Te-chung Issac Yang, Haowei Hsieh

Abstract

Digital signal processing techniques have been applied to analyze physiological signals for decades. Recent progresses in other fields, such as computer vision and machine learning, are attracting people to utilize such technologies for analyzing physiological signals. In this paper, recurrent neural network, often used in deep learning for time series signals, is applied to detect anomalies in heart sound. We successfully detected anomalies with 80% accuracy when augmenting the signals with other features.

1. Introduction

Physiological signals are good sources of information for physicians to analyze the health conditions of patients. Of the different types of physiological signals, heart sound and electrocardiography are the most common sources of information. Electrocardiography (ECG) is a process that records the electrical activity of the heart over a period of time using electrodes placed on the skin. ECG provides physiological signals related to the status of the heart. While medical equipments for recording, displaying, sharing and transferring ECG are usually used in hospitals and medical facilities, heart sound -- another type of cardiac signals, which can be observed with simpler equipments to perform similar functions, is still used by most primary care doctors.

1.1. Processing heart sound

Acoustic signal, the form of heart sound, has been studied heavily during the last few decades, thanks to various digital signal processing techniques. The general concept of acoustic signal processing is to apply a series of operations on one-dimensional signals called time series. For instance, speech recognition usually includes the following operations: digitization of analog signals, digital filtering, the cepstral transformation, the hidden Markov model, phonemes sequence generation, etc. However, in recent years, it has been shown that the accuracy of speech recognition can be improved by utilizing a different series of operations, most notably deep neural networks[1].

Another application of deep neural networks, which gained significant improvements during the last ten years, is image classification. In several well-recognized competitions among researches from either academia or industrials (such as ImageNet Large Scale Visual Recognition Competition (ILSVRC)[2] and PASCAL Visual Object classes (PASCAL VOC)[3]), deep neural networks that consist of tens of convolutional layers in neural networks of different topologies often outperform traditional, computer-vision-based technologies[4-7]. People now believe that deep neural network, when equipped with enough data and sufficient computation power, is capable of extracting features of higher complexities. As a result, those extracted features provide more insights on the finer details inside images than the manually crafted features, previously developed and commonly used in computer-vision application.

1.2. Using deep neural network

Since acoustic signals are one-dimensional and conceptually simpler than the more complex two-dimensional signals, like images, people also believe it would be possible to apply deep neural networks on acoustic signals and let the network extract features for classification. We follow such belief and adopt the state-of-the-art recommendation on image classification to build our own deep neural networks. With our network and training set provided by Physionet[8], we could achieve 80% overall accuracy with the revised scoring function. Based on experiences gained from other applications of deep neural networks, we believe that the results can be further improved by increasing the depth of the networks and by training the networks with more data in order to avoid overfitting.

2. Data preparation

2.1. Temporal segmentation

For each wave file, we performed the following transformation and converted it into a few samples. First,

we defined an observation window to be 4.8 seconds in temporal domain. We discarded any wave file that was shorter than 4.8 seconds. On each wave file, we applied the observation window with 50% overlap, and discarded the last window if it was shorter than 4.8 seconds. As such, a wave file of 15.9 seconds in length was converted into five samples whose temporal intervals were [0, 4.8], [2.4, 7.2], [4.8, 9.6], [7.2, 12.0] and [9.6, 14.4] seconds, respectively. The last window whose temporal interval was [12.0, 15.9] was shorter than 4.8 seconds and hence was discarded.

2.2. Labeling

These samples from previous conversion were labelled according to its original wave file. In the original labelling scheme, we chose to enforce with either "anomaly" or "normal". In the revised labelling scheme, an additional label, i.e. "noisy", was added. Note that we did not utilize segmented information about noisy intervals from signal quality files. The reason was due to the noisy duration not always being aligned with the observation windows. The misalignment could introduce mislabelling and thus lower the accuracy of ground truth.

2.3. Using training to extract augmented features

In addition to the above data, a few augmented features were included into the sample. The concept to include augmented features was inspired by ResNet[9]. In this paper, the author proposed convolutional neural networks whose learning parameters are the residual of original network. Therefore, the network was trained to find the residual weights, rather than the actual weight, for layers of convolutional neural networks. For example, if an original layer was $y = f(x)$, where x was the output of the previous layer. $F(x)$ is usually a matrix multiplication, where its elements, i.e. weights, were to be found during the training stage. For later discussion, we denote the matrix by M and assume $y = M * x$ for simplicity. The author of ResNet suggested that instead of directly finding elements of M through the training, the network can be rewritten as $y = x + H * x$. It is easy to see $M = H + I$, where I is the identity matrix, and H acts like the residual. Since the two expressions, $y = M * x$ and $y = x + H * x$, are equivalent, once $M = H + I$, the networks built from both expressions are mathematically equivalent.

However, the residual network proposed by the paper showed significant improvement over the original network in both accuracy and training speed. The author argued that the training became easier in the sense of finding optimal weights. Equipped with the better speed of the training stage in ResNet, the author increased the depth of the network in order to improve the classification power of the network. With sufficient amount of training data, the author showed the network outperformed any other network in image classification while the training speed was still within acceptable range.

The key point in this paper on deep learning neural networks for image classification suggests that even a network can be trained to learn weights and can classify images with great accuracy; its mathematically equivalent network may still learn faster and perform better classification. Being able to train the equivalent network loosens requirements on computation power and constraints on network depth.

We therefore added two main features in our data preparation stage. First, we performed windowed Discrete Fourier Transform (DFT) on each data sample. The window size was selected as 256 points, and 50% overlap between two consecutive windows. The magnitudes of the lower half of DFT coefficients were included in that data sample as the magnitudes of all DFT coefficients are symmetric. Second, the variance and the standard deviation of that window were also included. The selection of these two types of information was inspired by several facts. In digital signal processing, analysis in frequency-domain is common and has been proven useful. It also gives information invariant to temporal location. As we saw in 2.1, the temporal locations of observation windows to form data samples were chosen arbitrarily. In order to give the neural networks sets of information that were temporal location invariant, DFT was a good option.

Another reason to add DFT was that while the transform itself can be easily done in neural networks, as it was a matrix multiplication, magnitudes of coefficients could not easily be learned in the networks. To reduce computation requirements, such conversion was a preferred treatment during the data preparation stage.

The second set of augmented information, the variance and the standard deviation, was selected based on a similar concept. Also note that both the variance and the standard deviation were good estimates of loudness in acoustic wave files. They provided the neural networks with information similar to the envelopes of heart sound.

3. Neural networks

The commonly used neural network architecture of time series is recurrent neural network (RNN)[10]. Contrary to feed-forward neural network or convolutional neural network, RNN is able to 'memorize' history inside each data sample. Such capability is strongly preferred for extracting causal features in time series. The time interval between extracted causal features can vary. In other words, RNN is capable of finding two relevant events even if the time lag between their occurrences varies among data samples. In the sample codes provided by this competition, the envelopes of heart sound were extracted

with the help of the Hilbert transform and segmentation was performed by the hidden Markov model (HMM) and logistic regression [11][12]. As this approach suggested, the intervals of heartbeats in one wave file were considered as features and thus extracted. Since heart rate irregularity is an indicator of possible heart anomaly, that set of features was fed to the support vector machine in sample codes to detect anomalies. The RNN integrates those stages into one single network.

3.1. Detecting irregularity

In order to construct RNN for detecting such anomalies, we enforce a limitation on the minimum length of observation window, and use 4.8 seconds in our current design. If the window is too short such that it contains only two heart beats, the network will have no information to determine the irregularity of heart beats.

3.2. Detecting murmur

The other possible anomaly is murmur in heart sound. We believed a good feature to detect that type of acoustic signal was in frequency domain. However, RNN must know to look into murmur signals within certain temporal locations, namely around heartbeats. We augmented the data samples with DFT coefficient magnitudes to make it easier for RNN to jointly locate heartbeats and murmur in both temporal and frequency domains.

The first layer of our RNN is Gated-Recurrent-Units (GRU)[13] with 386 features. The next layer is a dropout layer[14], with dropout rate = 0.5. The third layer is another GRU, whose output is 8 features. The last layer is a fully connected layer with three label outputs. The first layer feeds its output back to its own during each timestep. The second GRU outputs its result at the end of the data sample. The number of timesteps in one data sample is 75.

The above hyper-parameters were chosen from the best result in cross-validation after grid search on different combinations of possible hyper-parameters. The topology was chosen based on the size of the dataset to avoid overfitting.

4. Training, validation and inference

The total number of data samples was about 420,000. One fifth of the data samples, randomly selected, were used for cross-validation. At least 2,000 epochs of training was done during grid search for hyper-parameters. The final hyper-parameter was used with 4,000 epochs of training. After the training, the model, including weight file and network file, was saved for inference stage. At the inference stage, the wave file was processed with the same data preparation procedure. Then the model was used to process each data sample to generate the inference results. A simple max argument voting scheme was used to select the final label for that wave file. That is, the most likely label among all data samples from one wave file was selected as the label of that wave file.

To submit results to the test server, we slightly modified the initialization step in inference to reduce setup time. It is likely the deep learning framework, i.e. Theano, needs to compile computation graph into GPU codes, which takes a long time at the beginning. We initiated the compilation process in setup.sh with a dummy wave file to save time.

The following figures show how the network learned in terms of accuracies and loss during training stage.

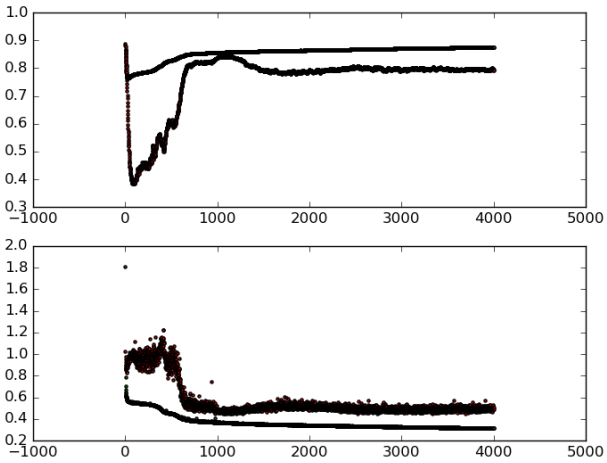

Figure 1: The upper subplot shows accuracies of training and validation sets. The lower subplot shows the loss function of those two sets.

5. Results

Our results achieved a top-10 ranking during phase one, with overall accuracy at 84%. With revised scoring, the overall accuracy dropped to 79%.

Several other attempts were made to improve the overall accuracy. While the accuracies from those attempts were not improved, one specific attempt, based on simplified three-layer one-dimensional convolutional network, increased specificity from 82% to 83%. Because convolutional neural network is well-studied and better supported in most deep learning frameworks compared to RNN, we believed it was possible to combine both networks to gain better performance. For example Caffe[15] added RNN support just two months ago.

6. Conclusion and future work

We proposed an end-to-end process to classify acoustic

physiological signals based on recurrent neural networks. To make the network easier to train, we augmented the data with features extracted from common digital-signal-processing techniques. The results showed a simple network was able to classify signals with around 80% accuracy.

While our results were not significantly better than other traditional approaches, the framework and technique we proposed can be easily extended when more data becomes available. Many researchers in deep learning also believe the deep learning techniques will outperform traditional methods when more data is available during training. To extend our network, we suggest knowledge transfer between networks -- the commonly used approach in convolutional neural network in image classification, object detection or segmentation. To be specific, a model trained with a particular data set can transfer its knowledge by exporting its learned weights. Another model on the same data set or same model on a different data set can load the entire or a part of the exported weights. The knowledge from the previous model is embedded into the weights and further training could add new knowledge into them.

References

[1] Hinton G, Deng L, Dahl GE, Mohamed A, Jaitly N, Senior A, et al. Deep neural networks for acoustic modeling in speech recognition. IEEE Signal Processing Magazine. 2012; 29(6): 82–97.

[2] Russakovsky O, Deng J, Su H, Krause J, Satheesh S, Ma S, et al. ImageNet large scale visual recognition challenge. International Journal of Computer Vision (IJCV). 2015; 115(3): 211-252. doi:10.1007/s11263-015-0816-y

[3] Everingham M, Van Gool L, Williams KI, Winn J, Zisserman A. The PASCAL visual object classes (VOC) challenge. International Journal of Computer Vision (IJCV). 2010; 88(2): 303-338.

[4] Krizhevsky A, Sutskever I, Hinton G. Imagenet classification with deep convolutional neural networks. Advances in Neural Information Processing Systems. 2012; 25: 1106–1114.

[5] Simonyan K, Zisserman A. Very deep convolutional networks for large-scale image recognition.

[6] Szegedy C, Liu W, Jia Y, Sermanet P, Reed S, Anguelov D, Erhan D, et al.. Going deeper with convolutions. Computer Vision and Pattern Recognition (CVPR). 2015; doi: 10.1109/CVPR.2015.7298594.

[7] Girshick R. Fast R-CNN. arXiv:1504.08083, 2015

[8] Liu C, Springer D, Li Q, Moody B, Juan RA, Chorro FJ, et al. An open access database for the evaluation of heart sound algorithms. Physiological Measurement. 2016; 37(9)

[9] He K, Zhang X, Ren S, Sun J. Deep Residual Learning for Image Recognition. arXiv:1512:03385. 2015.

[10] Connor JT, Martin RD, Atlas LE. Recurrent neural networks and robust time series prediction. IEEE Transactions on Neural Networks. 1994; 5(2): 240-254.

[11] Springer DB, Tarassenko L, Clifford GD. Logistic regression-hsmm-based heart sound segmentation. IEEE Transactions on Biomedical Engineering. 2016; 63(4): 822-832.

[12] Springer DB, Tarassenko L, Clifford GD. Support vector machine hidden semi-markov model-based heart sound segmentation. Computing in Cardiology. 2014.

[13] Cho K, Merrienboer B, Gulcehre C, Bahdanau D, Bougares F, et al. Learning phrase representations using RNN encoder-decoder for statistical machine translation. arXiv preprint arXiv:1406.1078, 2014.

[14] Srivastava N, Hinton G, Krizhevsky A, Sutskever I, Salakhutdinov R. Dropout: A simple way to prevent neural networks from overfitting. Journal of Machine Learning Research. 2014; 15: 1929-1958.

[15] Jia Y, Shelhamer E. Donahue J, Karayev S, Long J, Girshick R, et al. Caffe: Convolutional architecture for fast feature embedding. arXiv:1408.5093.

Address for correspondence.

Te-chung Isaac Yang
San Ramon, CA 94582, USA
tcyang@live.com

Haowei Hsieh
Lexington, MA 02421, USA
haoweih@gmail.com