Soonview algorithm for CPSC 2020 Challenge

2020-10-17

Wang Chi， Longbo Zhang

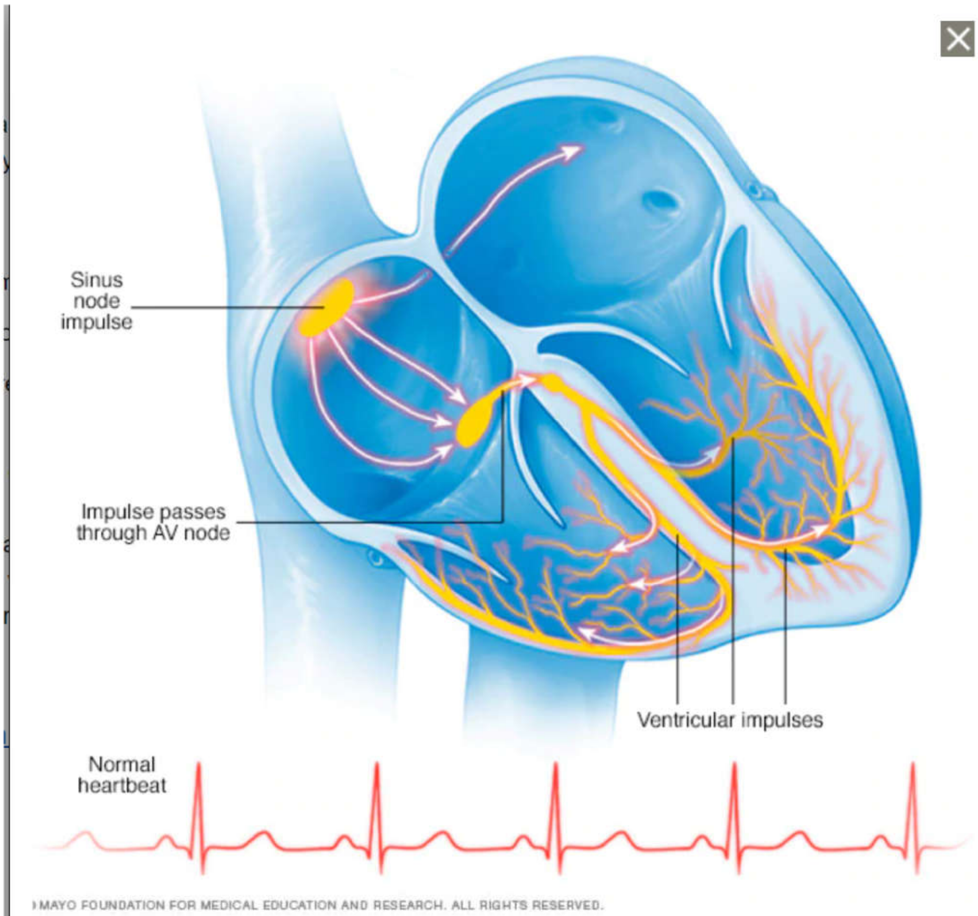
Shenzhen Soonview Technology Co., Ltd

Introduction：

The CSPC(China Physiological Signal Challenge) 2020 is developing algorithm to detect premature ventricular contraction (PVC) and supraventricular premature beat (SPB)。This document describe the design, implementation of our algorithm, and discuss possible future improvement.

Problem statement：

For normal heart beat, the electrical signal is flowing in the following way.



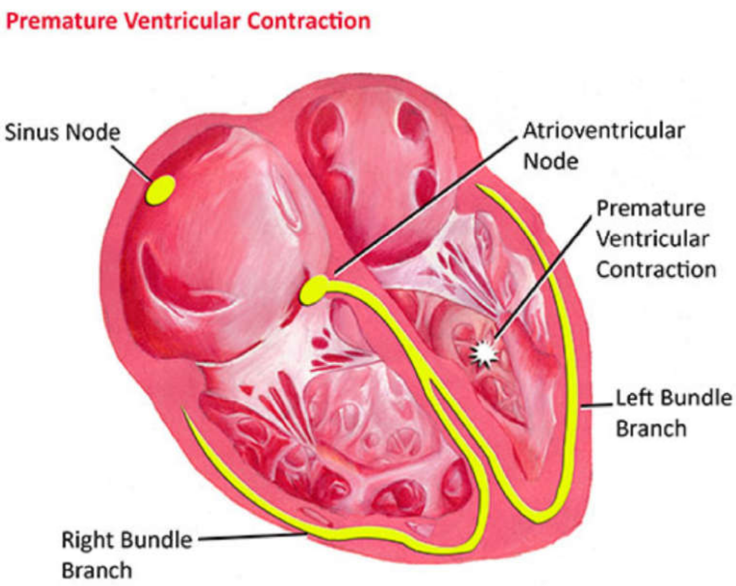
Normal heartbeat

In a normal heart rhythm, a tiny cluster of cells at the sinus node sends out an electrical signal. The signal then travels through the atria to the atrioventricular node and then passes into the ventricles, causing them to contract and pump out blood.

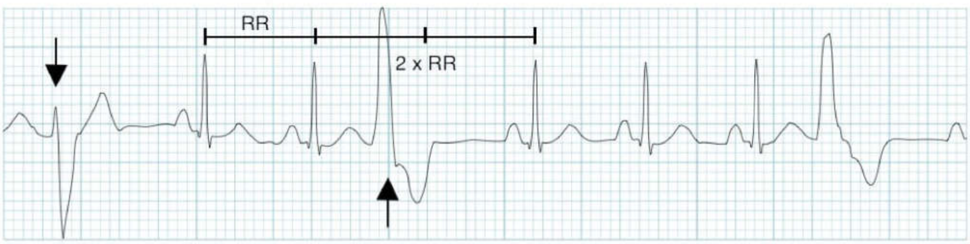
credit: https://www.mayoclinic.org/diseases-conditions/premature-ventricular-contractions/symptoms-causes/syc-20376757

According to Mayo Clinic, Premature ventricular contractions (PVCs) are extra heartbeats triggered by signal from lower chambers.

```
Premature ventricular contractions (PVCs) are extra heartbeats that begin in one
of your heart's two lower pumping chambers (ventricles). These extra beats
disrupt your regular heart rhythm, sometimes causing you to feel a fluttering or
a skipped beat in your chest.
```



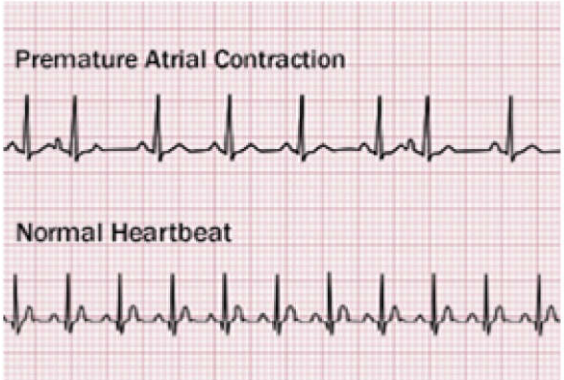
credit: https://pediatricheartspecialists.com/heart-education/18-arrhythmia/187-premature-ventricular-contractions



credit: https://litfl.com/premature-ventricular-complex-pvc-ecg-library/

supraventricular premature beat has the following definition (credit: https://www.uptodate.com/contents/supraventricular-premature-beats)

```
Supraventricular premature beats represent premature activation of the atria from
a site other than the sinus node and can originate from the atria or the
atrioventricular node (called junctional premature beats), though the vast
majority are atrial in origin. Premature atrial complexes (PACs; also referred to
a premature atrial beat, premature supraventricular complex, or premature
supraventricular beat) are triggered from the atrial myocardium in a variety of
situations and occur in a broad spectrum of the population. This includes
patients without structural heart disease and those with any form of cardiac
disease, independent of severity.
```

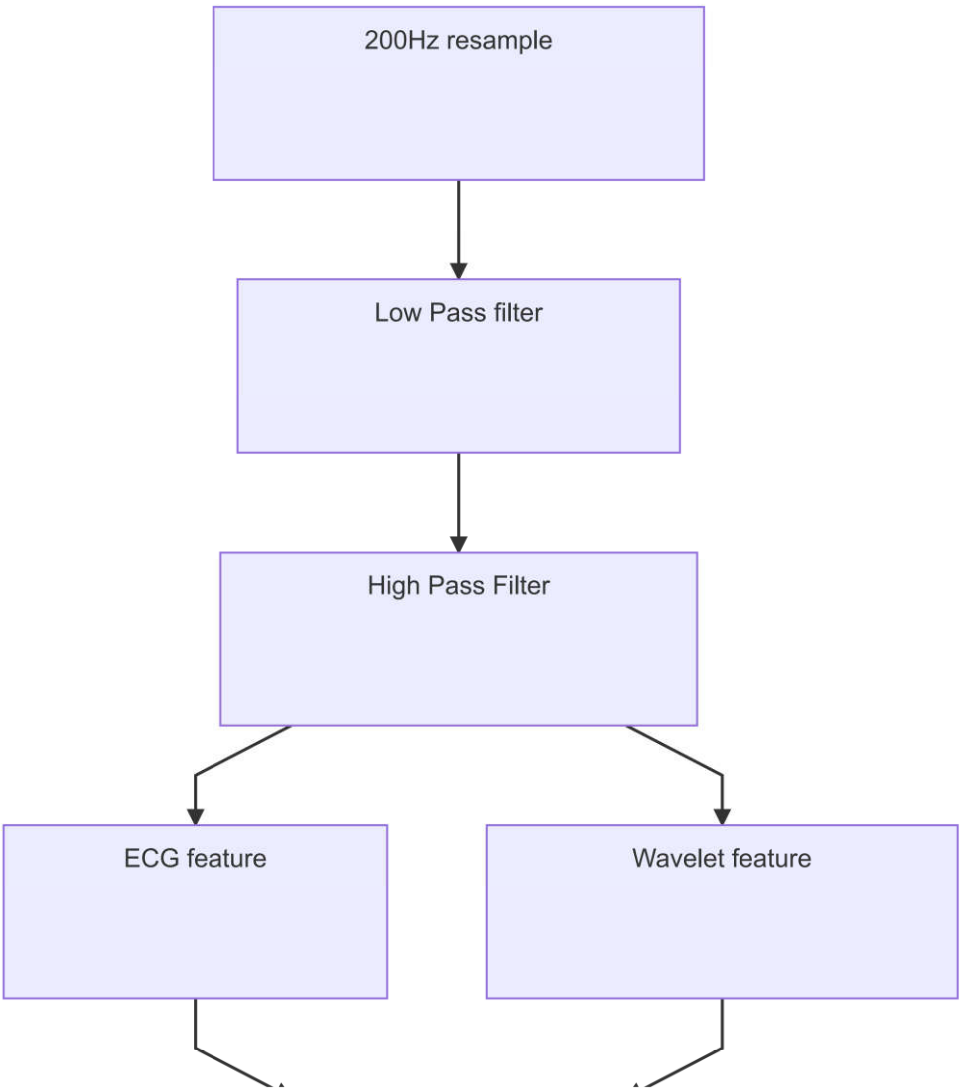


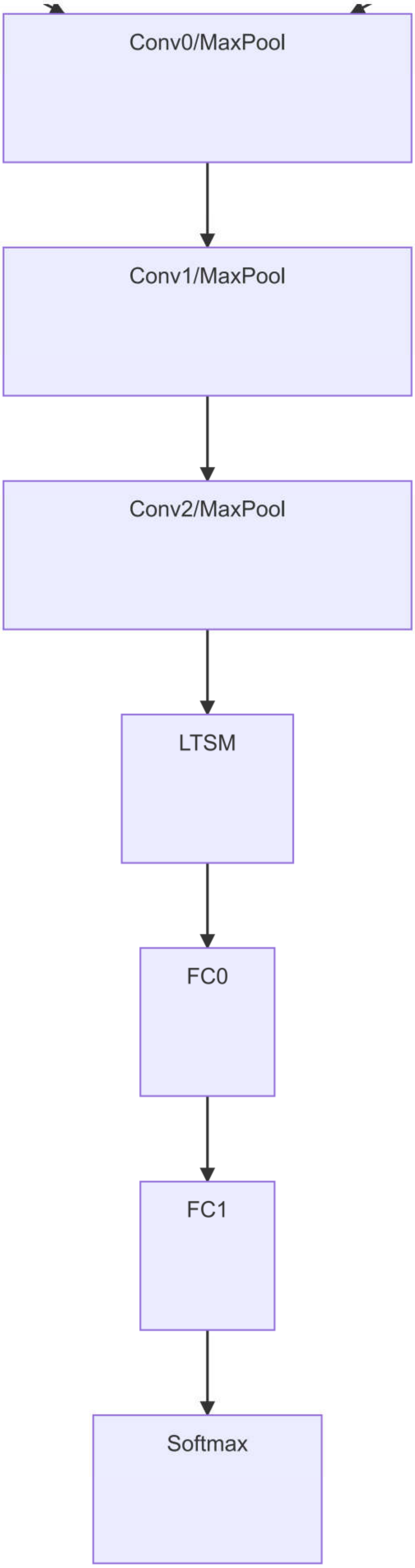
Algorithm design：

Traditional algorithm design normally has two steps. First does beat detection, then followed by beat classification. But for CPSC 2020, it will be difficult to use traditional design. The ECG data here is 24 hour dynamic signal-lead recordings usually with low signal quality and/or abnormal rhythm waveforms. It is really difficult to do reliable beat detection for noisy ECG data. The unreliable first stage beat detection will cause problem in the second stage beat classification.

We use end-to-end deep learning design here for beat classification, bypass the first beat detection stage.

The model is shown below:




Conv0/MaxPool
Conv1/MaxPool
Conv2/MaxPool
LTSM
FC0
FC1
Softmax

ECG feature: 300 ms sliding window ECG is used as input. Step size is 150 ms, aka there is 150 ms overlap between two 300ms sliding windows. We resample 400MHz original ECG record to 200Hz. ECG feature has 60 samples (0.3*200Hz = 60). This design converts beat detection and classification problem into a simple classification problem, avoids the potential risk of unreliable beat detection.

Wavelet feature: We also use wavelet analysis of ECG data as manually designed feature input. There are 37 features from wavelet analysis. ECG feature and wavelet feature are concatenated together to form the model input.

Label: The PVC and SPB labels are projected into 150 ms window as model target. 150 ms corresponds to the center of 300 ms sliding window.

object function: We use negative log likelihood loss with weight. The weight design is based on training data set PVC and SPB distribution. Since in the score function, SBP is weighted by 5, and PVC is weighted by 1, we multiply the distribution based weight by 5. The final weight used is (SPB, PVC)=(1700, 700), normalized to unit vector.

Steps for the training:

```
python preparebeat.py
```

This script read the data in Matlab MAT format, doing 400Hz->200Hz resampling, then run through lowpass filter and high pass filter to remove high frequency noise and base line shift. It then splits the data into 300 ms data segments with 150 ms overlap, calculates ECG feature and wavelet features, store them as training input. It also reads the labels, and converts them into targets for the 300ms model inputs. All the input and object are stored in LMDB database.

```
python trainbeat.py --dbname cspc2020data.db
```

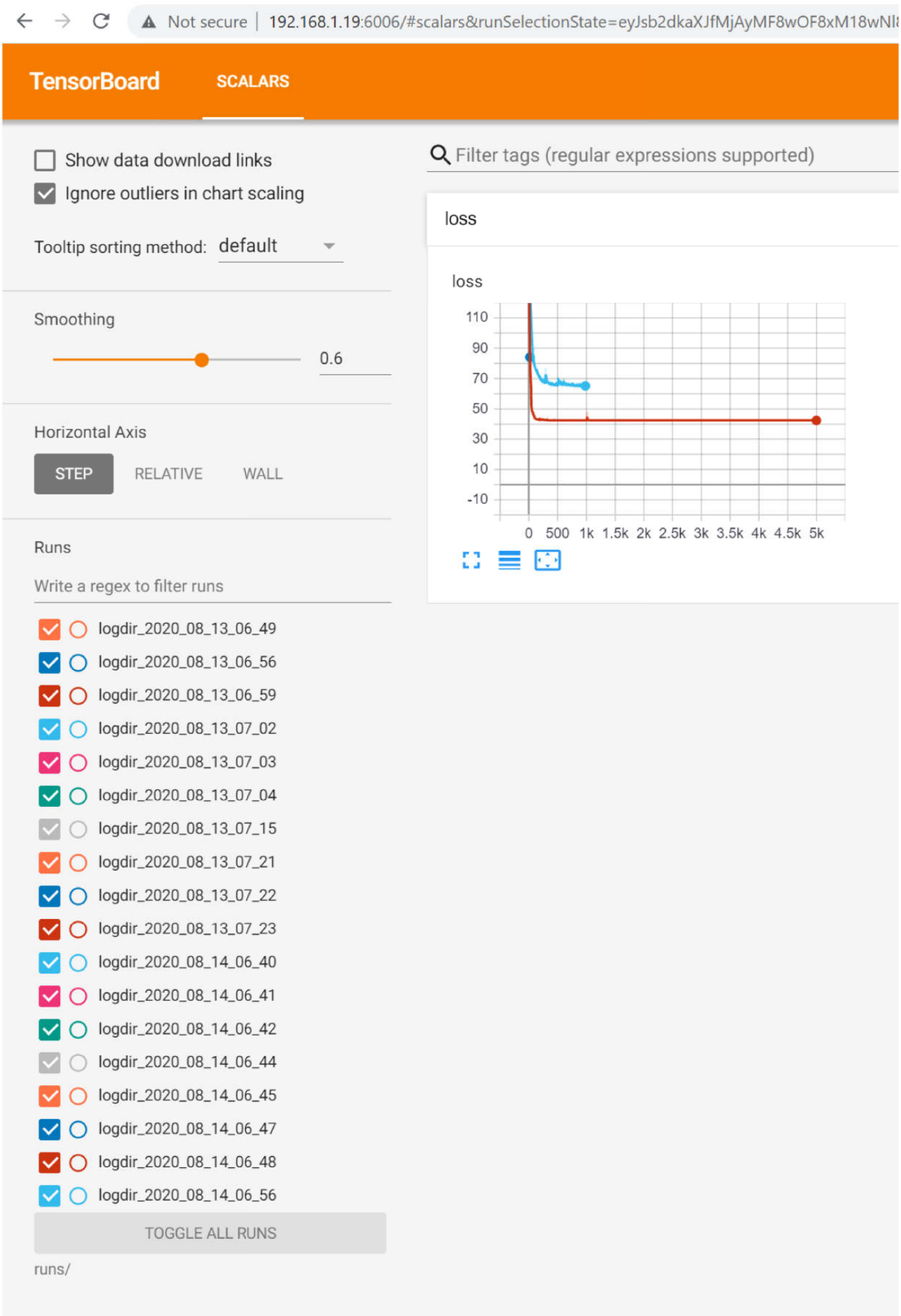
The above command is used to do the training. It requires Nvidia GPU, CUDA and pytorch. There are options to pick epoch, resume training, or adjust negative log likelihood loss with weight.

```
usage: trainbeat.py [-h] [--epoch EPOCH] [--resume RESUME] [--dbname DBNAME]
                    [--sweight SWEIGHT] [--vweight VWEIGHT]

optional arguments:
  -h, --help         show this help message and exit
  --epoch EPOCH      number of epoch, default: 5000
  --resume RESUME    resume training from hidden states and state dict,
                     default: None
  --dbname DBNAME    train database, default: beat.db
  --sweight SWEIGHT  weight given to S classification, default: 1700
  --vweight VWEIGHT  weight given to V classification, default: 700.0
```

Tensorboard writer writes training logs into runs folder, we can use Tensorboard and web browser to check training progress.

```
(base) @aibox:~/sw/challenge$ tensorboard --logdir runs/ --host 0.0.0.0
2020-10-17 21:48:29.430367: I
tensorflow/stream_executor/platform/default/dso_loader.cc:48] Successfully opened
dynamic library libcudart.so.10.1
TensorBoard 2.3.0 at http://0.0.0.0:6006/ (Press CTRL+C to quit)
```



Potential improvement：

- For commercial product, we should spend time improving the raw ECG signal. Based on results from third party database, a better design is AI combined with traditional beat detection approach. Sliding window should be beat based, not time based. Different beat can be aligned with R wave. Then classification algorithm can be implemented with AI.
- With R wave detection and heart rate detection, we can bring in more classical features into AI classification input. For example, experiments have show heart rate values around current beat are very useful in classification. The width of QRS complex is also a very useful feature.
- For deployment on phone or portable device, speed and power consumption is a major design goal. For future work, we can also think on implementation that not depends on floating point math. For example, with pytorch and tensorflow, we can use quantification aware training. https://pytorch.org/docs/stable/quantization.html. https://www.tensorflow.org/model_optimization/guide/quantization/training。