Rays* "FOR EXCELLENCE

Deep Learning

N.B.: The following is a research diary written by Hugo Bergand intends to record the progress and problems encountered by the author during RAYS – For Excellence – 2019.

NOVEL METHOD ON EXISTING DATASETS

Introduction

> This project is, as of today, not very clear in its purpose.

During the cardiac cycle, the heart firstly generates the electrical activity and then the electrical activity causes atrial and ventricular contractions. This in turn forces blood between the chambers of the heart and around the body. The opening and closure of the heart valves is associated with accelerations-decelerations of blood, giving rise to vibrations of the entire cardiac structure (the heart sounds and murmurs). These vibrations are audible at the chest wall, and listening for specific heart sounds can give an indication of the health of the heart. The phonocardiogram (PCG) is the graphical representation of a heart sound recording. Figure illustrates a short section of a PCG recording.

The objective of the challenge is to create a model is able to correctly discriminate between the two classes given just the PCG recordings. Challenges include:

- Data is subject to temporal variations due to variations in the heart rate.
- Inter-patient differences make difficult a learn a model that generalizes well across patients.
- Differences introduced by heterogeneity in the collection of the recordings can render a classifier trained on one population useless when applied to another

I Dataset and Preprocessing

> In most Machine Learning research projects you will be using some kind of samples from a Dataset to learn a model. Thus it is extremely important that you carefully describe the dataset and why you believe is a good dataset for the project and what type of preprocessing are you going to apply

Early approaches failed to build a reliable model due to lack of a large enough data set, so this challenge provides the largest dataset to this day. Heart sound recordings were sourced from several contributors around the world, collected at either a clinical or nonclinical environment, from both healthy subjects and pathological patients. The Challenge training set consists of five databases (A through E) containing a total of 3,126 heart sound recordings, lasting from 5 seconds to just over 120 seconds.

A main problem found when working with these recordings is the strong similarity between the records

coming from the same population as well as the strong class imbalance of roughly 6:1 of Normal to Abnormal.

Table I.1 summarizes the sizes of the different populations as well as their class imbalance

	A	N	S	A/S
a	292	117	409	0.714
b	104	386	490	0.212
c	24	7	31	0.774
d	28	27	55	0.509
e	183	1958	2141	0.085
	631	2495	3126	0.202

(a) Training Set

	A'	N'	S'	A'/S'	S'/S
a	40	40	80	0.5	0.20
b	49	49	98	0.5	0.20
c	4	3	7	0.57	0.23
d	5	5	10	0.5	0.18
e	53	53	106	0.5	0.05
	151	150	301	0.50	0.10

(b) Validation Set

Table I.1: Population properties $A \equiv$ Abnormal, $N \equiv$ Normal, $S \equiv A + N$

I.1 Preprocessing

There is a high heterogeneity since it was compiled in different environments with diverse systems and devices. The DTW affinity between representative heartbeats is completely biased if we do not apply any kind of preprocessing.

To prevent this a simple zero mean unit variance normalization approach is used to get closer distances. Nevertheless with a reasonable $\sigma = 10^{2.5}$ we can note that the population distances are still there except less noticeable.

$$x_i' \leftarrow \frac{x_i - \bar{x}}{\sigma} \tag{I.1}$$

$$\bar{x} = \frac{1}{N}\sum_{i=1}^{N} x_i \qquad \sigma^2 = \frac{1}{N}\sum_{i=1}^{N}(x_i - \bar{x}) \tag{I.2}$$

All the provided records were sampled at $f_s = 2\,\text{kHz}$. The segmentation algorithm resamples them[1] to $f_s' = 1\,\text{kHz}$.

Strike-through

Often you will be wrong on your assumptions, but do not throw them away completely, just cross them out in case you need them later using the `\sout` command and it will look like ~~this~~ or `\soutthick` command and it will look like ~~this~~

~~Medoid computation is perfomed at $f_s'' = f_s'/5 = 200\,\text{kHz}$ to speed computation. Simple analysis was performed to check that the features extracted from these 200Hz-medoids were approximately the same as the ones extracted from the 1khz-medoids.~~

[1] Resampling = Low Pass Filter + Downsampling by M. The filter will have $\omega_c = \pi/N$ to prevent aliasing

~~Downsampling can be seen as a problem of information loss in the frequency spectrum. If the frequency content $f > f_s/(2N)$ is mostly empty for N which we downsample by said N we will only be losing information in that range.~~

TODOs

You will often have pending tasks that you need to track. This research journal allows you to include both high and low priority todos that will be summarized in a list at the end of the file. Use the commands `\hightodo` and `\lowtodo`, including a date is recommended for tracking purposes. Note: define your `userId` in the preamble

HB: 2016-05-23 : Compute the error between mode-downsampled segmentation state vectors at 1000 Hz and state vectors computed at 400 Hz. This needs to be performed to check that the segmentation algorithm is not overfitted to 1000Hz. If error is significant, retrain segmentation at 400 Hz or use 1000 just for segmentation (if Matlab 2016a improvements are true there should be no problem)

HB: 2016-05-27 : Do DRYRUN with new Matlab 2016a and check the segmentation quota

II Methods

II.1 Algorithms

Sometimes the most straightforward way to explain a procedure is just to give it in a algorithmic format, it takes a little time but it will force you to go thorugh the steps and you will most likely be able to reuse it on you paper.
Note: You will need to have `\ALGORITHMStrue` in the preamble to enable algorithms

Algorithm 1 Euclid's algorithm

procedure EUCLID(a, b) ▷ The g.c.d. of a and b
  $r \leftarrow a \bmod b$
  while $r \neq 0$ **do** ▷ We have the answer if r is 0
    $a \leftarrow b$
    $b \leftarrow r$
    $r \leftarrow a \bmod b$
  return b ▷ The gcd is b

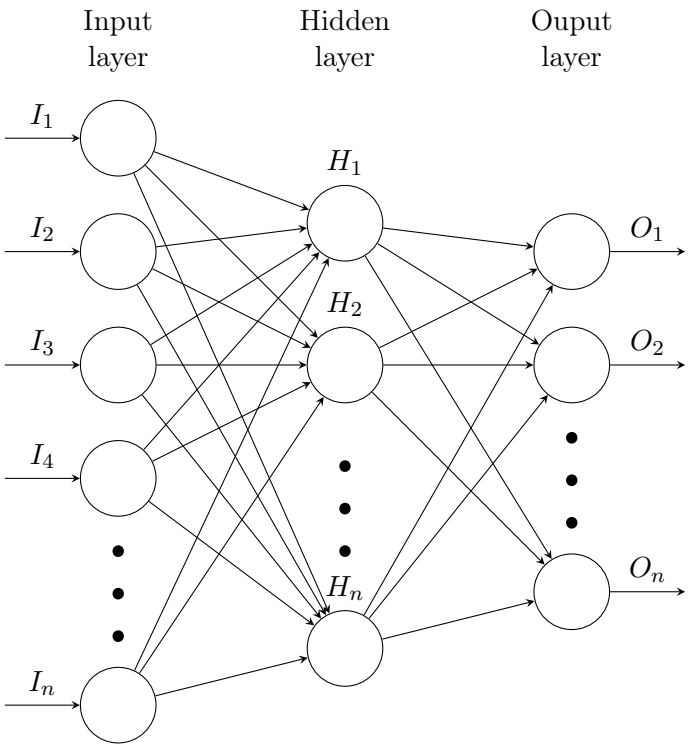
II.2 Code

If the algorithm is to vague and you feel like you need the source code you can also insert it. You can put LaTeX code inside by using `<@ @>` delimiters and highlight it with `<| |>` delimiters
Note: you will need to have `\LISTINGStrue` in the preamble.

```
def DTW_distance(s1, s2):
    """
    Function to compute the Dynamic Time Warping in Python between two signals
    """
    DTW={}

    for i in range(len(s1)):
        DTW[(i, -1)] = float('inf') # By default ∞
    for i in range(len(s2)):
        DTW[(-1, i)] = float('inf') # By default ∞
    DTW[(-1, -1)] = 0

    for i in range(len(s1)):
        for j in range(len(s2)):
            dist= (s1[i]-s2[j])**2


    return sqrt(DTW[len(s1)-1, len(s2)-1])
```

II.3 Diagrams

For simple diagrams I highly recommend learning TiKZ, you will be drawing the diagrams in pure LaTeXwhich has a steep learning curve but once you get used to it, it can be quite easy to display and do `for` loops to draw multiples line at once.
Note: you will need to have `\TiKZtrue` in the preamble



However, sometimes you will need more complicated diagrams (or maybe you do not like TiKZ, in that case I recommend a vector drawing tool such as Inkscape which allows LaTeX embedding)

III Results

III.1 Figures

In general the best way to visualize your results will be some figures, I recommend Python's matplotlib for generating them or R's ggplot2.

III.2 Tables

LaTeX booktab environments are really good to showcase and track your results, however they can get fairly messy. My suggestion is to generate them via Python automatically and store the results in either a plain text file or a spreadsheet (there are packages to read spreadsheets with Python)

$\sigma \setminus \tau$	0	1	2	3	4	5	6	7	8
0.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0
0.2	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0
0.4	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0
0.6	98.6	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0
0.8	84.7	99.5	100.0	100.0	100.0	100.0	100.0	100.0	100.0
1.0	28.1	98.3	99.9	100.0	100.0	100.0	100.0	100.0	100.0
1.2	1.3	88.7	99.4	99.9	99.8	99.9	100.0	99.9	100.0
1.4	0.0	57.1	96.2	99.3	99.0	99.3	99.4	99.8	99.7
1.6	0.0	18.6	81.2	93.0	93.7	94.8	95.6	92.3	93.3
1.8	0.0	2.4	42.8	67.0	70.1	72.1	69.0	69.1	68.6
2.0	0.0	0.1	9.0	23.1	24.5	26.9	28.2	27.3	27.3
t(ms)	27.92	40.23	77.30	157.27	252.05	342.18	381.46	399.85	413.72

Table III.1: Performance of the algorithm for 128-bit key and with multiple readings per key

A Resources

It is a good idea to record sources that explain concepts or provide tools so the research is both better documented and if someone has to continue with it, there is enough supporting documentation.

- Quick read in DTW and Keogh Lower Bounding

  http://alexminnaar.com/time-series-classification-and-clustering-with-python.html

  http://nbviewer.jupyter.org/github/alexminnaar/time-series-classification-and-clustering/blob/master/Time%20Series%20Classification%20and%20Clustering.ipynb

- Parallelizing DTW – Good article on making a parallel version of DTW. Uses Keogh lower bound not as a linear approximation but as a pruning device.

  https://www.andrew.cmu.edu/user/mmohta/15418Project/finalreport.html

- Deep Learning
  - Intro to LSTM
    https://colah.github.io/posts/2015-08-Understanding-LSTMs
  - Intro to CNN
    https://colah.github.io/posts/2014-07-Conv-Nets-Modular/
  - Why are LSTMs are so useful, impressive result in character pattern and syntax learning
    https://karpathy.github.io/2015/05/21/rnn-effectiveness/

B References

Do not forget to cite the papers that you are using in your research, this way the **Previous Work** part in your paper will be infinitely easier to write when the time comes.

References

C TO DO

Here you will have all your TODOs grouped with anchor links to the parts of the document where they are. Really handy if you do not know where to continue with your project.

Todo list