***09.2019***

Pierwsza stycznosc z danymi – analiza kodu napisanego przez Marcina – zapoznanie z PyTorchem aby zrozumiec jego kod. Czytanie tutoriali z PyTorcha, uruchamianie przetwarzania danych, uczenia modelu GRU w PyTorchu napisanego przez Marcina.

Ogarnianie frameworka Analysta stworzonego przez promotora.

Analiza przykladowych szeregow czasowych – zaobserwowano podobienstwo sampli w ktorych pacjent zemdlal i takich w ktorych nei zemdlal, wniosek: ciezko odroznic jedne od drugich. Niektore wygladaja nawet na zle olabelowane, duzy spadek cisnienia krwi/tetna nie powoduej zemdlenia? Moze blad maszyny?

Analiza modelu marcina - widac ze dzieli wszystkie szeregi na probki 300 elementowe oznaczone 1 albo 0 i uczy na tym siec, po czym model predykuje liczbe z zakresu 0,1 dla kazdej probki testowanego szeregu, czyli tak jakby wylicza prawdopodobienstwo. Tylko na poczatku kazdy szereg wyglada tak samo wiec przewidziec zaslabniecie mozna raczej na chwile przed faktycznym zaslabnieciem, bo dla wiekszosci sampli w szeregu dopoki parametry nie swiruja mamy bardzo niskie prawdopodobienstwo zaslabniecia.

***10.2019***

Zastanawianie sie nad czescia jezykowa – pomysl stworzenia modelu jezykowego ktory zawiera nazwy lekow i oblicza odleglosci nazw lekow od slowa ‘mdlec’. Zaobserwowano problem – bardzo malo danych tekstowych (moze kilkadziesiat probek).

***11.2019***

Proby wyjasnienia dlaczego podejmujemy decyzje o tym ze dany szereg czasowy jest przykladem zemdlenia – autoenkoder wariacyjny (VAE). Visualization with T-SNE and PCA, rather unsuccessful.

***12.2019***

Each timeseries is interpolated, with outliers removed, smoothed and stored in separate CSV file.

Pipeline for loading the data from text files into one data matrix was created.

More intense utilization of LSTM networks. Whole pipeline to launch grid search was created, including:

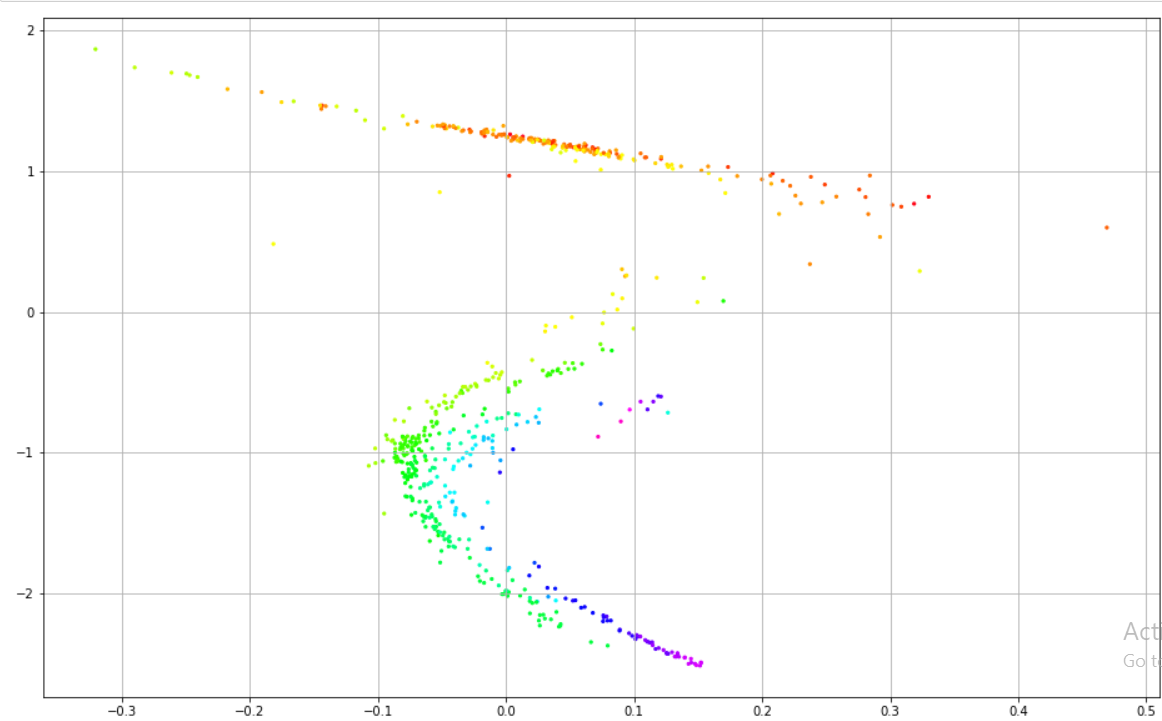
* Preprocessing:
  + MinMaxScaling,
  + cutting first 40% samples from each timeseries,
* Training with the following hyperparameters:
  + Number of LSTM layers: [1, 2, 3, 4]
  + Number of cells in each LSTM layers: [1, 16, 64, 256]
  + Number of time steps considered as 1 sample: [32, 64, 128]
  + Number of neurons in fully connected layer: [5, 50]
  + Number of epochs: [1, 3, 5]
  + Batch size: [128]

Some of the models learned quite successfully. Responses to all validation samples are saved so as to compare them and choose the most promising one.

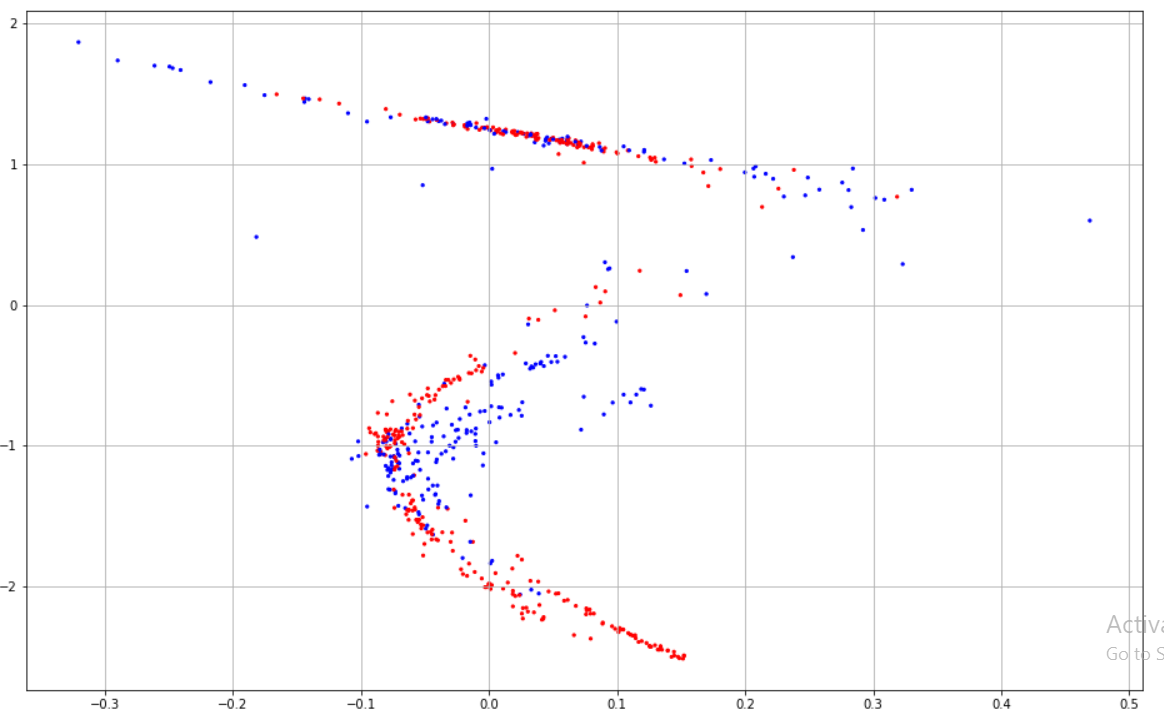
***5-13.04.2020***

Variational Recurrent Autoencoder tested on simple data, proved to work properly. In the beginning technique proposed by the NN owner was applied eg. reduce to 20 dims and then use PCA/TSNE to visualize, but after all similar results were obtained with directly reducing dimensionality to 2. Applied to my HR/BP data with direct dim reduction to 2 with poor results. VRAE learns the mean BP/HR and visualize on 2D instead of learning sudden raises or drops. Surely more benchmarking is necessary to try to teach better model.

Mean visualized (64 timesteps/40epochs/default config):



Labels(64 timesteps /40epochs/default config)



***19.04.2020***

Reading about pretrained recurrent networks. Some info about TimeNet, ConvTimeNet (papers) but nets aren’t available publicly. Reading about downsampling, function to downsample everything to 200 without much loss of the trend created. To be tested.

***20.04.2020***

Tested undersampling. It is not appropriate technique for this type of a problem, because to measure patient health parameters in real time it is necessary to rely on small time intervals. Single information about classification the whole timeseries undersampled from 5000 to 200 points is useless and inaccurate.