Project Proposal

Group members: Evian Liu, Lisa Kim, Shiyu Dou, Zhihao Xu

As COVID-19 pandemic started, patients with trouble breathing increased and mechanical ventilation became more in need. However, mechanical ventilation is a clinical-intensive procedure and with a limited number of clinicians, it is difficult to match the rising demand. Therefore, we aim to simulate a ventilator connected to a sedated patient’s lung. A successful simulated ventilator that performs better than the existing PID controller would be able to increase the accessibility of the ventilator treatments and also reduce the cost of developing new methods for controlling mechanical ventilators. The task of our project is to predict the airway pressure in the respiratory circuit during the inspiratory phase of each breath.

We obtained this dataset from Kaggle competition which is sponsored by Google Brain and Princeton University. (<https://www.kaggle.com/c/ventilator-pressure-prediction/data?select=test.csv>)

The data variables this project deals with are: R, C, u\_in, u\_out, time\_step, pressure. R and C variables describe the lung attributes of an individual person so each sample of the patient would have these variables fixed as the above figure shows; R indicates how restricted the airway is, and C represents the lung compliance (how easily the lung expands).

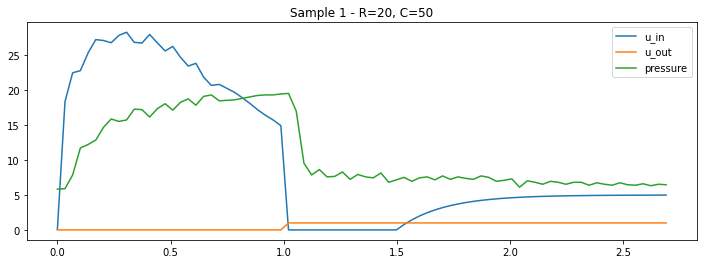


Figure 1, one example of simulation input

Given the C and R variables in the inspiratory phase (u\_out = 0), we are using the time series of the u\_in values to predict the airway pressure at each timestamp of the breathing period. More specifically, u\_in is a continuous variable from 0 to 100 representing the percentage the inspiratory solenoid valve is open to let air into the lung (i.e., 0 is completely closed and no air is let in and 100 is completely open). u\_out is a binary variable that indicates the state of the inhalation vs exhalation where u\_out = 0 means it’s in the inspiratory phase, while u\_out = 1 means it is in the expiratory phase. This project will use u\_in, R and C (possibly u\_out) as the input of our neural network, and the output we aim to predict is pressure.

In order to predict unknown respiratory patterns of patients, the neural network has to capture features of simulated ventilator behaviors. Thus fine-tuning hyperparameters and network structure design would be the main challenge. First challenge lies in deciding the best training data. It is intuitive to subset u\_out = 0 test data because the exploratory phase always happens after the inspiratory phase. It is also possible that the historical data collected during the inspiratory phase could improve the predictive power of our model. Therefore, we may also explore whether to incorporate the cases with u\_out = 1 in the training stage of our model. However, incorporating cases with u\_out = 1 poses computational challenges because it doubles the size of our training dataset.

We will use LSTM with denoise autoencoder and attention to capture the complex dynamics within the respiration sequences. Rather than dimension reduction, autoencoders will denoise input sequences. We choose not to use the bottleneck layer/output of the encoder as an input for regression, but pass an LSTM classifier with the weights of the encoder model directly. To encode each time sequence simulation input, we will remember the hidden states of information across long input sequences and output the number of timesteps and the hidden dimension. We will train the embedding by its pressure prediction in the training dataset. Experimenting different numbers of layers of LSTM is another option. If we have time, we will also explore whether the attention mechanism improves the fixed-weights LSTM algorithm.

Qualitatively, we will examine how closely the prediction outputs match up with the actual test dataset in timed plots. Quantitatively, we will measure the mean absolute error across all breaths in the test set. In particular, it is the error between the predicted and actual pressure values during the inspiratory phase of each breath. And we can compare such scores with other traditional methods, such as linear regression and other machine learning methods.

Project Proposal Ver.1

Group members: Evian Liu, Lisa Kim, Shiyu Dou, Zhihao Xu

As COVID-19 pandemic started, patients with trouble breathing increased and mechanical ventilation became more in need. However, mechanical ventilation is a clinical-intensive procedure and with a limited number of clinicians, it is difficult to match the rising demand. Therefore, we aim to simulate a ventilator connected to a sedated patient’s lung. A successful simulated ventilator that performs better than the existing PID controller would be able to increase the accessibility of the ventilator treatments and also reduce the cost of developing new methods for controlling mechanical ventilators. Therefore, the goal of our project is to use numerous time series of breaths to predict the airway pressure in the respiratory circuit during the breath as opposed to forecasting the future values using the past data.

In order to predict unknown respiratory patterns of patients, the neural network has to capture features of simulated ventilator behaviors. Thus fine-tuning hyperparameters would be the main challenge. It is also critical to design the network structure of input data reconstruction with spatial attention and sparse encoding. Besides, it would be challenging to visualize such timed data patterns. Furthermore, we want to find ways to evaluate feature importance as the traditional approach we use for tree-based methods are not applicable under the LSTM framework.

We have obtained this dataset from Kaggle competition which is sponsored by Google Brain and Princeton University. (<https://www.kaggle.com/c/ventilator-pressure-prediction/data?select=test.csv>)

We will use LSTM-autoencoder with attentions for this multivariate time series. We will design a spatial attention mechanism in RNN with denoising and sparse Autoencoders.

Qualitatively, we will examine how closely the prediction outputs match up with the actual test dataset in timed plots. Quantitatively, we will measure the mean absolute error across all breaths

in the test set. In particular, it is the error between the predicted and actual pressure values during the inspiratory phase of each breath. And we can compare such scores with other traditional methods, such as linear regression, moving average, and etc.

**Topic Brainstorming**

* LSTM: <https://paperswithcode.com/paper/extreme-volatility-prediction-in-stock-market>
* Datasets: <https://www.v7labs.com/blog/best-free-datasets-for-machine-learning#computer-vision>
* Time Series: LSTM algorithm

<https://machinelearningmastery.com/lstm-autoencoders/>

**Example of LSTM Autoencoders:**

* <https://www.kaggle.com/dimitreoliveira/time-series-forecasting-with-lstm-autoencoders>
* <https://www.kaggle.com/nicholasjhana/energy-consumption-generation-prices-and-weather>
* <https://www.kaggle.com/dimitriosroussis/electricity-price-forecasting-with-dnns-eda/notebook>

**Multivariate time series datasets**

Organized by data field: <https://github.com/awesomedata/awesome-public-datasets>

<https://machinelearningmastery.com/time-series-datasets-for-machine-learning/>

* EEG Eye State Dataset: whether their eyes were open or closed

<https://archive.ics.uci.edu/ml/datasets/EEG+Eye+State>

* Occupancy Detection - room occupancy

<https://archive.ics.uci.edu/ml/datasets/Occupancy+Detection+>

* Ozone Level

<https://archive.ics.uci.edu/ml/datasets/Ozone+Level+Detection>

* Time series data

<https://github.com/xephonhq/awesome-time-series-database>

* Bitcoin <https://archive.ics.uci.edu/ml/datasets/BitcoinHeistRansomwareAddressDataset>
* Google Brain - Ventilator <https://www.kaggle.com/c/ventilator-pressure-prediction/data?select=test.csv>

**Mechanism**

* LSTM-autoencoder with attentions for multivariate time series (<https://github.com/JulesBelveze/time-series-autoencoder>)

(<https://github.com/Seanny123/da-rnn>)

(<https://github.com/Zhenye-Na/DA-RNN>)