Intermediate Report

Katarina Kuchuk, Xinze Li

November 19, 2021

Introduction

The competition Google Brain - Ventilator Pressure Prediction is an example of using ML-based methods to simulate the ventilator-lung system.
The goal of the ventilator is to regulate the pressure sensor to measurements to follow the target waveform. The main problem to be solved in the real world is to build a robust controller of ventilation. One of the possible approaches is first to use the ML model to train a differentiable simulator of the lung and then to train a controller based on this simulator. This competition focuses on the first step of this process: training a simulator of the lung.

Data Description

The dataset contains following features:
Parameters of ventilator:

- `u_in`: the control input for the inspiratory solenoid valve, a float value from 0-100
- `u_out`: the control input for the exploratory solenoid valve, either 0 or 1.

Parameters of lung:

- `R`: the change in pressure per change in flow (air volume per time)
- `C`: the change in volume per change in pressure

Other Parameters:

- `id`: globally-unique time step identifier across an entire file
- `breath_id`: globally-unique time step for breaths
- `time_step`: the actual time stamp
- `pressure`: target

The dataset contains 75450 breaths (unique `breath_id`), each breath has 80 timestamps. Here is a visualization of a sample breath:

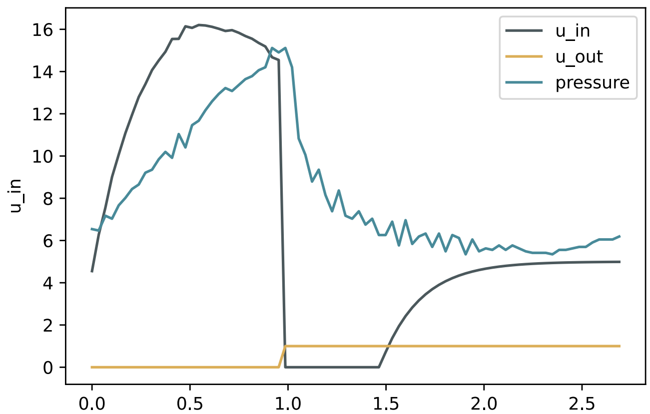

Figure 1: Visualization of main features in one `breath_id`

Progress

[24.9 - 03.10] build framework with basic models

We build a framework of our project, including pipeline, metric, models, config file. The code is organized to have easy-to-use interfaces to make us more easily focus on feature selection and model selection. And we use config file to launch each experiment to record and organize our experiment results. The framework has two versions, PyTorch and TensorFlow, corresponding to two branches in github repo, they are similar in architecture.

[03.10 – 30.10] model development, error analysis

This problem is somehow like a regression version of Part-of-speech tagging. So we have tried LSTM, TCN, and transformer models, since they have already proven results in these fields. The best result we got using LSTM, so we have mainly focused on this model. The following are some of our conclusions:

1. Training different models for different type of lungs can not improve the results.

2. We made error analysis and found that there are some errors that comes from the incorrect prediction of the first pressure in a breath (Figure 1).

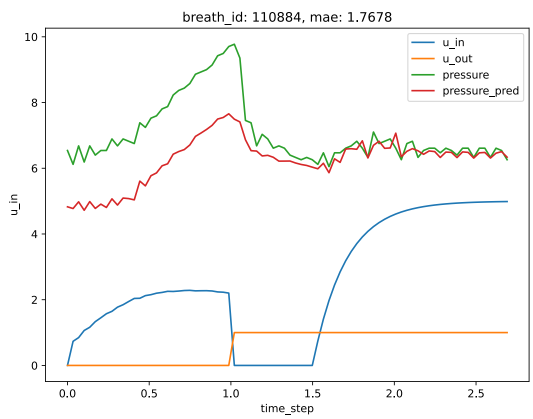

Figure 2: A typical situation when model gives bad prediction

The whole `breath_id` is continuous, but some of them were taken out as the test data. For those data whose previous `breath_id` existed in the train dataset, we can calculate the difference between the last pressure in a breath and the first pressure in the next breath. The pressure difference has approximately a normal distribution with large variance (Figure 3). Figure 4 shows two typical situations of the pressure difference.

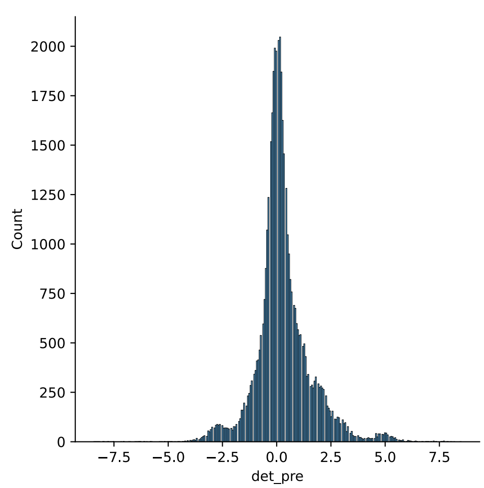

Figure 3: Distribution of difference between the last pressure in a breath and the first pressure in the next breath

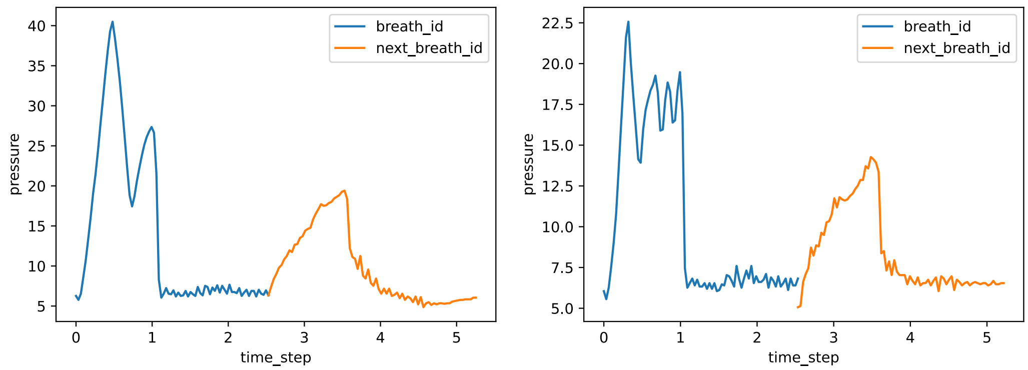

Figure 4: On the left is the ideal situation, the pressure difference is very small, on the right is a case showing the large variance

3. We have trained a model with an additional cheating feature: the first pressure in a breath, and it resulted in a 10% decline in MAE. Based on that result, we tried different ways to predict the first pressure in a breath more precisely, including training a model using the feature in current `breath_id` and previous `breath_id` to predict the first pressure in current `breath_is`, but they could not outperform the baseline model. We finally tried to first train a model with the cheating feature, and then use the first pressure predicted by the baseline model as the cheating feature for prediction, and it resulted in a 3% decline in MAE.

Future work

The competition ended half a month ago, but it allows late submission. We have noticed that some of the top-ranked teams that have used a combination of LSTM and transformer got a really high rank, and because of that, we are going to investigate how they have achieved this. At the same time, it is worth noting that most models which include BiLSTM can contribute to the ranking, but can not be used in the real world due to the use of future information. We are going to find some top models and make an error analysis to find about how they perform in the typical error (the first pressure in a `breath_id` Figure 2). Maybe they can show a pattern rather than noise which can explain the pressure differences' normal distribution (Figure 3), which is the key of improving the result.