Sequence-to-sequence Approaches on Ventilator Pressure Predictions

Shiyu Dou*, Lisa Kim**, Evian Liu*, Zhihao Xu*
*Department of Statistics & Data Science**
*School of Public Health***
Yale University
shiyu.dou@yale.edu, lisa.kim@yale.edu, evian.liu@yale.edu, zhihao.xu@yale.edu

***Abstract*—With rising demand of Pressure Support Ventilation for patients with trouble breathing, the project employs various sequence-to-sequence models to predict ventilator pressures associated with heterogeneous patients' lung types. The transformer model gives the best model performance due to its superiority in capturing oscillation patterns compared with LSTM based models.**

***Index Terms*—seq2seq, LSTM, transformer, autoencoder, ventilator**

I. Introduction & Motivation

As COVID-19 pandemic started, patients with trouble breathing increased and mechanical ventilation became more in need. The role of mechanical ventilation is to help patients who have trouble breathing on their own. Despite its significance, mechanical ventilation is a clinical-intensive procedure as the respiratory therapists or MDs have to monitor patients' ventilation and physiological conditions frequently to maintain airway pressure and prevent leakage of secretions into the lungs [1]. As patients with different lung attributes respond differently to the ventilator, it makes the monitoring process more challenging. 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 entire breathing phase, both inspiratory and expiratory, using these features: measure of airway restriction (R), lung compliance (C), percentage the inspiratory solenoid valve is open to let air into the lung (u_in), and indicator value of inhalation and exhalation (u_out). The most successful model will be able to accurately predict the machine airway pressure at different time stamps given input u_in where the machine has control over, but also considering the heterogeneity of patients lung type which is captured by variables R and C.

II. Background & Related Work

This project is inspired by previous applications of deep long-short term memory models (DLSTM) on multivariate forecasting time series datasets. Sagheer et.al [2] proposed a DLSTM architecture that overcomes the limitations of shallow RNN and LSTM in univariate time series forecasting. Alternatively, the LSTM-based stacked autoencoder (LSTM-SAE) consists of 3 LSTM-AE blocks in the stack [2]. Each LSTM encoder layer is passed into the next block and trained to learn from original inputs. All three LSTM encoder layers are used to construct a final 3-layer DLSTM.

Sequence-to-sequence learning has been widely used in speech recognition and translation since 10 years ago. The addition of the attention mechanism largely improved information extraction and model performances. The Listen, Attend and Spell (LAS) is a neural network that transcribes speech to characters using a pyramidal RNN encoder listener and an attention RNN decoder [3].

It is surprising to see in the previous investigation of the dataset that most work focuses heavily on feature engineering in order to turn this many-to-many problem into a many-to-one problem in order to apply techniques such as the vanilla LSTM framework. This might be due to the popularity of using the seq2seq model (many-to-many LSTM, transformers) to improve machine translation accuracy. Therefore, we would like to apply the seq2seq model in this dataset without heavily relying on feature engineering.

III. Model

In this project, a variety of seq2seq models are applied to investigate time series data structures and predict the airway pressures. We implement various LSTM based autoencoder structures, and a transformer model. All the best performing models are achieved by fine-tuning the network architecture and parameters, including number and dimension of layers, learning rate, optimizer, activation function, batch size, dropout layer, number of epochs, etc. The input vector representation is centered and scaled based on percentiles before being passed into the model to ensure model-robustness and to avoid gradient explosion. We adopt early-stopping mechanism to prevent overfitting when necessary. In addition, Adam optimizer with gradient clipping added a final layer of protection against gradient explosion. The performance is evaluated by the mean absolute error between original and predicted values.

A. The Vanilla LSTM

RNN, consisting of internal states and short-term memories, is popular for modeling sequential data. Even better, the LSTM improves upon RNN by using a conveyor belt to send the past information directly into the future to deal with vanishing gradient problems and better learning long-term dependencies. We begin our implementation using 3 layers of LSTM to obtain a hidden representation of our input data. In this way, we learn the linear or non-linear projections from the original high-dimensional space in this vanilla LSTM model. We then send our learnt hidden representation into a dense layer to obtain our prediction of pressure. Note that the vanilla LSTM we implement in this subsection is not a seq2seq based model, so we will use it as a benchmark to compare its performance with the seq2seq models in the next few subsections. The demonstration of the model architecture is given in the next subsection of LSTM-AE.

B. The Seq2seq LSTM Autoencoder (LSTM-AE)

Autoencoder is a type of neural network that is trained to replicate its input to its output. By restricting the size of the bottleneck to be smaller than that of the input layer, one forces the autoencoder to capture the most salient features that can represent our input data. Extending on the previous vanilla LSTM with only the encoder and a 1D bottleneck, we implement a complete encoder-decoder LSTM model and compute prediction from the decoder output. Although LSTM intends to improve the long term memory, it can still forget information in the past if the input sequence is long. We thus use a bidirectional LSTM (Bi-LSTM) to make our architecture have longer memory. Note that only the encoder block uses Bi-LSTM, and we use single directional LSTM in the decoder block. Since we have a large number of observations, we stacked our LSTM by using 3 encoder Bi-LSTMs and 3 decoder LSTM. To mimic autoencoder structures, the dimensions of the hidden representation vectors follow a bow tie shape. The model architecture is given in the diagram below.

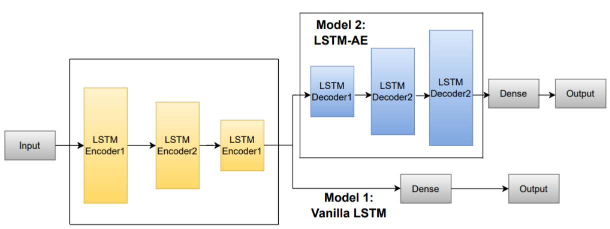

Fig. 1. Vanilla LSTM & LSTM-AE architecture

C. The LSTM Stacked Autoencoder (LSTM-SAE)

Deep learning revolves around the idea that a deep hierarchical architecture can be exponentially more efficient at representation than the shallow ones [4]. Sagheer et. al extended the LSTM-AE architecture to achieve a deeper hierarchical representation of our time-series data LSTM-SAE. [5] Our seq2seq stacked LSTM model has 3 layers of encoder LSTMs and 3 layers of decoder LSTMs. All 3 encoder LSTMs have the same structure. Different from LSTM-AE in the previous subsection, our autoencoders are stacked on each other. Concretely, the i-th encoder ($i \in \{1, 2, 3\}$) outputs hidden state of all the timesteps (H_i), the hidden state of the last time step (h_i), the last cell state of the last time step (c_i). There are respectively 3 LSTM decoders. The first decoder takes input c_1 and h_1 from the first encoder and outputs H_1^*, the second decoder takes input H_1^* and c_2 and h_2 from the second encoders. The third decoders outputs $H3^*$ using input H_2^* and c_3 and h_3. The output is sent to a time-distributed dense layer to generate the final prediction. The detailed architecture of our LSTM-SAE is given below.

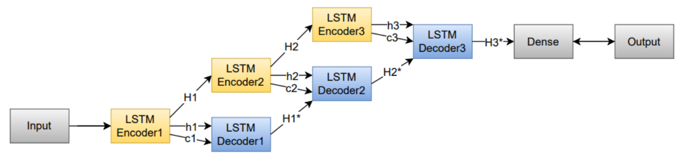

Fig. 2. LSTM-SAE architecture

D. The Transformer

One disadvantage of using LSTM based seq2seq models is that the hidden representation learnt by the encoder block (i.e hidden state and converybor belt) is still incapable of remembering a longer sequence. Attention mechanism [6] was first proposed to improve the performance of seq2seq models. The attention mechanism has two major advantages: (i) The seq2seq model does not forget source input. Different from the sequential embedding in LSTM models, the entire time-series input is embedded simultaneously, which incorporates an additional vector encoding of positional information in the embedded vector representation. This will further improve the long term memory capacity compared with Bi-LSTM. (ii) With attention, we would know which part of the inputs to focus on when predicting the output of a given timestamp. Point (ii) is particularly beneficial since our input data has two phases: inspiratory and expiratory. Their structure can be fundamentally different and the attention mechanism will be able to take that into account. In this section, we implemented a transformer model from paper 'Attention is All you Need' [7]. This architecture is built upon the self-attention mechanisms that outperform the classic seq2seq LSTM with attention structures, and is considered a state-of-the-art technique in the machine translation literature. The key component in the transformer architecture is the multi-head self-attention layer. The self-attention layer in speech recognition and translational tasks encodes the contextual information between words and sentences. For our data, it encodes the positional relationship between $breath\_ID$ and lung attributes R and C. a more detailed account of our network structure can be found in Vaswani et.al [7]).

IV. Empirical Results

A. Dataset Description

We obtained this dataset from Kaggle competition which is sponsored by Google Brain and Princeton University [8]. Variables related to one patient of lung type $R = 20$, $C = 50$ is given below.

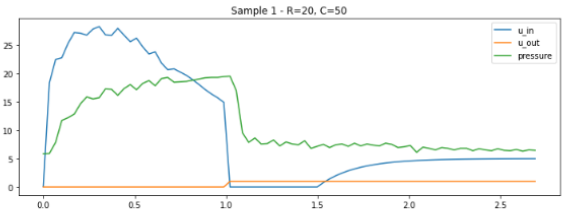

Fig. 3. Time series plot of variables u_in, u_out, $pressure$ of sample 1

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. In addition to C and R variables, we used the time series of the u_in and u_out 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.

B. Feature Engineering

Sequential observations on the control inputs for the inspiratory solenoid valve (u_in) and the exploratory solenoid valve (u_out) are separated by time stamps and identified by $breath\_ID$. To remove long-term structural patterns in the time-series data, feature engineering was performed on the input space. Lag function loops through u_in and u_out and calculates additional columns of lagged u_in and u_out on the timestamp and $breath\_ID$ to remove any linear or nonlinear temporal dependencies. Because the two lung attributes, R and C, are categorical variables, we performed one-hot encoding to create a new set of dummy variables that equals the number of categories in these two variables.

C. Model Results

- **Vanilla LSTM:** The best performing autoencoder was achieved with Adam optimizer at 0.001 learning rate on 128 batchsize. With the hyperbolic tangent activation function, the vanilla LSTM contains 2 hierarchical LSTMs and dense layers. The model starts out with a wide LSTM layer with 150 neurons and the last dense layer provides a single vector of pressure prediction. Training requires a mean squared error loss function, which gives the lowest test MAE of 0.53.
- **LSTM-AE:** The complete LSTM-AE consists of a 3-layer bidirectional encoder, a low-dimensional bottleneck layer and a 3-layer decoder. The best performing autoencoder was achieved with the hyperparameter setup as the Vanilla LSTM model. The encoder starts out with a wide LSTM layer with 500 neurons, and narrows down hierarchically to a bottleneck with width of 20. Then we experiment with the addition of drop-out layers to the AE architecture to prevent data points memorization. Training requires a mean squared error loss function, which gives the lowest test MAE of 0.38.
- **LSTM-SAE:** The seq2seq stacked LSTM model achieves a test prediction MAE score of 0.33.
- **Transformer:** The transformer model returns a test MAE score of 0.22. It achieves an improvement of 20%.

We are also interested in the model performance given different lung types. The performances are shown in the table below.

TABLE I: Test MAE of different lung types using different methods

R	C	LSTM-AE	LSTM-SAE	Transformer
5	10	0.318	0.262	0.179
	20	0.274	0.24	0.177
	50	0.298	0.275	0.226
20	10	0.325	0.287	0.194
	20	0.328	0.331	0.227
	50	0.371	0.321	0.263
50	10	0.401	0.310	0.176
	20	0.531	0.430	0.280
	50	0.549	0.439	0.283

Below is a plot of patient 1 from the testing sample and see the difference in performance across different method. All 3 seq2seq models are able to capture the general trend of the pressure output, the difference in terms of performance is resulted from their ability to capture periodicity (i.e periodic oscillation).

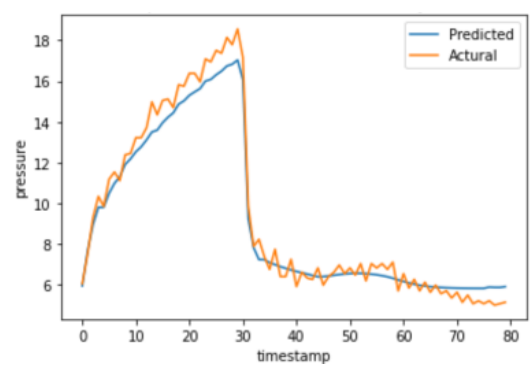

Fig. 4. Vanilla LSTM Fit of Sample Patient

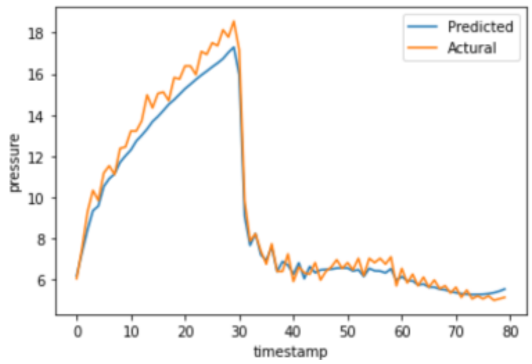

Fig. 5. LSTM-AE Fit of Sample Patient

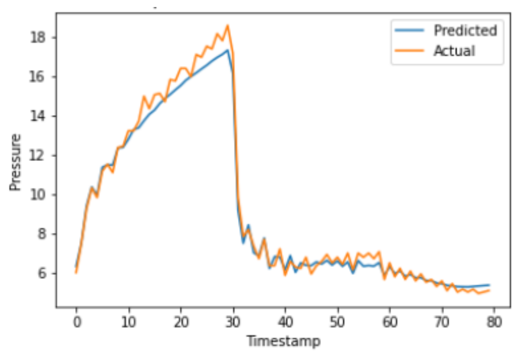

Fig. 6. LSTM-SAE Fit of Sample Patient

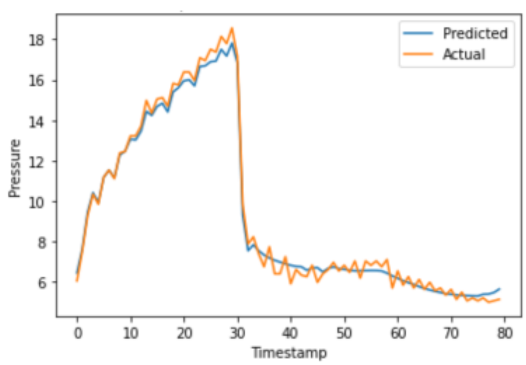

Fig. 7. Transformer Fit of Sample Patient

V. Conclusions & Future work

The project starts with a basic multi-layer LSTM model that experiments with feature engineering and hyperparameter tuning. Then it involves seq2seq models: LSTM-AE, LSTM-SAE and transformer. Both Seq2seq LSTM models and transformers are able to capture the general trend of the pressure output, but transformer is superior in capturing periodicity. Transformer achieves the best model performance as it is the only model among 3 that roughly captures the periodicity pattern, while LSTM based mode fails significantly. This is also shown in the section above that we find that the performance of all our seq2seq models compromises in some lung types. In particular, it is more likely to observe periodicity of u_in and $pressure$ in those lung types and our model does not perform well in those situations. This is consistent with the results of Liu et. al (2020) that demonstrated experimentally that the standard activations functions, such as ReLU, tanh, sigmoid, along with their variants, all fail to learn to extrapolate simple periodic functions. For instance, this can be shown in the sample 1 in the training set (see the plot above). In the future, we would like to better incorporate periodicity in our model training. We consider using discrete Fourier transformation (DFT) in the spectral analysis literature to capture the dominant frequencies of oscillation. The DFT transforms the data from time domain to frequency domain. The j-th Fourier transform is defined as

$$J_j = \sum_{t=1}^{T} x_t \exp(i\omega_j t) \quad j = 0, ..., T-1$$

where ω_j is the j-th Fourier frequency defined as $\omega_j = \frac{2\pi j}{T}$. In our case $T = 80$. If we can incorporate those Fourier analysis into feature engineering, we might see an improved performance of our seq2seq model.

Specialization

Since this project involves 4 models, we divided up the work among the four group members evenly. We subdivided into 2 groups for better communication and brainstorming. In particular, Kim and Liu are responsible for the LSTM and LSTM-AE models. Dou and Xu are responsible for the LSTM-SAE and Transformer models. Team members fully involved with model architecture design, hyperparameter tuning, visualization and related sections of this document.

Acknowledgment

We thank professor Smita Krishnaswamy for creating this fantastic deep learning course and showing us the frontier knowledge in this field. We also appreciate all teaching fellows/ ULAs for their help and tutoring throughout this semester.

References

[1] “Monitoring during mechanical ventilation,” AARC, 19-Aug-2021. [Online]. Available: https://www.aarc.org/an21-monitoring-during-mechanical-ventilation/. [Accessed: 28-Apr-2022].

[2] A. Sagheer and M. Kotb, “Time series forecasting of Petroleum Production using Deep LSTM recurrent networks,” Neurocomputing, vol. 323, pp. 203–213, 2019.

[3] W. Chan, N. Jaitly, Q. Le, and O. Vinyals, “Listen, attend and spell: A neural network for large vocabulary conversational speech recognition,” 2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2016.

[4] R. Pascanu, C. Gulcehre, K. Cho, and Y. Bengio, ”How to construct deep recurrent neural networks”, ICLR, 2014.

[5] A. Sagheer and M. Kotb, “Unsupervised pre-training of a deep LSTM-based stacked Autoencoder for multivariate time series forecasting problems,” Scientific Reports, vol. 9, no. 1, 2019.

[6] D. Bahdanau, K. Cho, and Y. Bengio, ”Neural machine translation by jointly learning to align and translate”, ICLR, 2015.

[7] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. Gomez, Ł. Kaiser, and I. Polosukhin. ”Attention is all you need.” Advances in neural information processing systems 30, 2017.

[8] "Google Brain - ventilator pressure prediction," Kaggle. [Online]. Available: https://www.kaggle.com/c/ventilator-pressure-prediction/data?select=test.csv. [Accessed: 28-Apr-2022].