Project Proposal: Reproducing Temporal Pointwise Convolutional Networks for Length of Stay Prediction in the Intensive Care Unit

Jacob Hennig

Introduction

Problem Statement

In the paper "Temporal Pointwise Convolutional Networks for Length of Stay Prediction in the Intensive Care Unit", the authors Rocheteau, Liò, and Hyland (2021) address the challenge of predicting the Length of Stay (LoS) for ICU patients. This challenge poses a crucial problem regarding managing resources like bed availability and for improving patient care. Methods that are commonly used, such as RNNs and Transformers in particular, struggle with processing Electronic Health Record (EHR) data due to its irregular sampling, skewness, and missing values. This makes it difficult for them to effectively capture long range temporal dependencies. For addressing these issues, the authors proposed using another model known as a Temporal Pointwise Convolutional (TPC) network. This model applies both temporal and pointwise convolutions directly to each time step of the patient data. This approach also helps to capture local temporal patterns and dependencies between variables more efficiently, leading to improved accuracy and scalability for LoS prediction compared to existing methods.

Methodology

Specific Approach

The paper introduces a TPC network to predict ICU LoS. Unlike RNNs and Transformers, which often struggle with irregular and skewed EHR data, the TPC network uses a different approach to handle these. Essentially instead of processing data in a step by step way, TPC applies pointwise convolutions directly to the temporal data. This avoids the need for sequential processing, resulting in faster computations while still effectively capturing local temporal dependencies. The model uses a kernel size of 5 to balance capturing precise temporal features without adding too much computational overhead. Also, the TPC's architecture is designed to handle long range dependencies, avoiding the vanishing gradient issues that can occur in recurrent architectures.

The model is trained to minimize the Mean Squared Logarithmic Error (MSLE) to address the positive skew in LoS data. Evaluation metrics include the Mean Absolute Deviation (MAD) of the remaining LoS, Mean Squared Error (MSE), R^2, and Cohen's Kappa Score. The authors compare their model against commonly used Long-Short Term Memory (LSTM) networks and Transformer models on the eICU and MIMIC-IV datasets. The results from this showed that TPC consistently outperforms RNNs and LSTMs in the prediction accuracy and scalability for ICU LoS prediction.

Novelty & Hypotheses

The main novelty of this paper follows the common theme that this project proposal has discussed, the TPC network itself. Its improvements over RNNs, LSTMs, Transformers, etc. due to their computational inefficiencies along with its own unique detailed approach, prove it to be a superior model for EHR data and long range dependencies.

The primary hypothesis that the authors made in this paper is that the specific combination of temporal and pointwise convolutions in the TPC network are able to capture local temporal dependencies better than RNNs. With this then leading to improved accuracy and scalability. Their results supported the hypothesis with TPC network outperforming RNNs and LSTMs in many key evaluation metrics, again including MAD, MSE, and Cohne's Kappa. So the hypothesis and novelty of this paper both tie in to the fact that the TPC network model is designed to be more computationally efficient, while offering better prediction accuracy for ICU LoS than baseline models.

Ablations/Extensions

Potential ways to improve their method:

1. Assess the impact of the mortality prediction side task by removing the mortality prediction side task to see how it affects LoS prediction accuracy.
2. Testing different temporal convolution kernel sizes (3, 5, 7) to analyze how they impact capturing local or broader patterns.
3. Conducting external validation using a holdout set from MIMIC-IV to assess generalizability.

The core of the hypothesis behind the TPC network is solid. It directly addresses the limitations of baseline models for handling long range dependencies and irregular EHR sampling. Plus then the paper's results show that the TPC network outperforms models like RNN-based models in both their accuracy and scalability. All of this information makes the hypothesis well supported.

Data Access and Implementation Details

Data Access

The eICU Collaborative Research Database and MIMIC-IV are accessible via PhysioNet (https://physionet.org). In order to access this I would have to obtain credentialed access and complete the data use agreements. For doing this, I will follow the "Getting MIMIC Access" project instructions.

Codebase

The study uses the eICU Collaborative Research Database and MIMIC-IV dataset. The authors have made their code publicly available on GitHub at: https://github.com/EmmaRocheteau/TPC-LoS-prediction. This repository includes aspects of the model implementation, scripts for data loading, data preprocessing, and the training and evaluation code used in their experiments.

Computational Feasibility

Computation for this seems to be feasible using GPU resources, such as what is available in Google Colab. Plus it seems that the TPC network model allows for training within a few hours, which'll likely be manageable within Colab's session limits, as I can save progress and restart where needed. Then additionally with the code all being well documented, and having plenty of relevant information included on the GitHub, I should have many resources to assist me if need be.

Use of Existing Code

I plan to use the existing code that is provided by the authors in the GitHub repository, as it looks well documented and seems to allow for consistent replication of the original approach.

References

Rocheteau, E.; Liò, P.; and Hyland, S. 2021. Temporal Pointwise Convolutional Networks for Length of Stay Prediction in the Intensive Care Unit. In *Proceedings of the ACM Conference on Health, Inference, and Learning (CHIL '21)*. Association for Computing Machinery. DOI: https://doi.org/10.1145/3450439.3451860.