Hospital Emergency Visit Prediction from Patient Claims Data

Balaji Sathyamurthy BALAJIS2@ILLINOIS.EDU
Ramasamy Ramanathan RR13@ILLINOIS.EDU
Gowri Shankar Ramanan GSR2@ILLINOIS.EDU
Raja Mukherjee SM56@ILLINOIS.EDU

Abstract – Emergency claims prediction is very critical for the healthcare organization for both payers and hospitals. It can save cost for the health insurance companies, hospitals and patient and can provide the patient with an efficient care management. Emergency claims are high dollar claims and most of the times, the reason behind the claim is due to an underlying disease pattern. By identifying the patient disease pattern, who will be likely going to ER, the health insurance company and alert their primary care physician so that patient can be diagnosed for critical conditions and can reduce the number of ER visits. This is mutually beneficial to both the healthcare insurance companies since it saves a lot of high dollar claims and as well as for the patient by avoiding them from going to the ER.

We utilize claims data to identify patterns with the primary objective of flagging claim sequences which will help determine the propensity of a patient to go for an emergency visit. We pass visit sequences through embedding layer, run it through GRU and get results through a fully connected layer. Our model performance compares favorably against baseline Random Forest model and LSTM. Our approach closely follows these papers: Doctor AI (Choi et al. 2016) and DLP (Yogesh et al. 2019).

For implementation details, please refer to:
https://github.com/bsathyamur/ERVisit-GRU-MultiheadedAttn

Introduction

Just one trip to the ER could cost you as much as a mortgage payment, with the average visit costing about $2000 - $5000. But costs can rise fast -- if you need emergency surgery or must be admitted! And even if the ER is in your insurance company network, some of the doctors who treat you in the ER may not be in the insurance provider network, which will drive costs up even higher. A recent study found as many as one in five ER bills carries out of provider network charges!

Avoidable visits to emergency departments are among the major factors contributing to rising healthcare costs, and of the roughly 27 million annual ED visits by privately insured patients, about two thirds are avoidable, finds a new brief from UnitedHealth Group.

Part of the problem is that primary care services rendered by hospital EDs come with substantially higher price tags than in primary care settings.

In fact, the average cost of treating 10 common treatable conditions at a hospital ED is $2,032, which is more than $1,800 higher than in primary care situations. The ED cost is 12 times higher than at a physician's office ($167) and 10 times higher than at an urgent care center ($193).

The 10 treatable conditions examined in the brief were. bronchitis, cough, dizziness, flu, headache, low back pain, nausea, sore throat, strep throat, and upper respiratory infection. Higher costs at hospital EDs are partially driven by a couple of different factors. For one, there are hospital facility fees, which increase the cost of an average ED visit by $1,069. There are lab, pathology, and radiology services, which average $335 at an ED. That is about 10 times more costly than at a physician office, where the same services average about $31.The money adds up. Of the 27 million annual hospital ED visits by privately insured patients in the U.S., 18 million were deemed avoidable, meaning those patients can be treated safely and effectively in high-quality, low-cost primary care settings.

When those 18 million avoidable ED visits are multiplied by an estimated $1,800 cost reduction per visit, that translates to a **$32 billion** annual savings in US healthcare system opportunity if care is diverted to a primary care setting. This has inspired us to solve the problem by identifying the patients early and making them not to visit ER and go to Primary care settings would save a lot of money for both hospitals and patients. The quality of the care in ER would also improve and the scheduling of the staff in the ER department can be done much efficiently knowing which seasons and when you will have more patients and what kind of care would be required.

Electronic Patient Health Record (EHR) data has vital records as well as visit sequences, so they help in understanding future patient visits or diseases by their construct. Models use enriched data i.e., combination of claims data, demographic data, images to amplify the signals. While predicting future visit data, the aim has been to use sequences of events i.e., visits over time, and learn their patterns. This in turn helps to predict the next emergency visit. The challenge here is that the data is inherently sparse and noisy, due to patient's irregular visits, incomplete/ absence of information etc.

In this paper, our goal is to predict the patient's likelihood of an emergency visit in near future (binary classification) based on the status of multiple diagnoses in the form of medical codes as inputs, found in the sequences of patient claims data having multiple severity levels. Deep learning techniques such as Recurrent Neural Networks (RNN) are necessary to correctly model the dependencies of diagnoses in large claims data while reasonably interpret the prediction results.

We propose an attention based RNN model to monitor patient's longitudinal health information from sequences of diagnoses found in historical claims data. First, we use an RNN to memorize all the information from historical visits, and then attention mechanisms to measure visit importance. We train the model to learn from the diagnosis sequences and focus on the prediction of the ER visit. Recurrent neural networks (RNNs), Long Short-Term Memory (LSTM) and Gated Recurrent Units (GRU), have achieved stat-of-the-art performance in handling long-term dependencies and nonlinear dynamics. Taking advantage of the capability of RNN in memorizing historical records, multiple recent models based on RNNs are employed for deriving accurate and robust representations of patient visits. This work will result in building a much useful application for hospitals and health care foundations to monitor disease progression in patients leading to emergency visits which will help in early treatment and better care for patients in an efficient way. This helps physicians in identifying patients from the cohort whose disease state is more likely to end up in Emergency visit which helps them to suggest precautions, lifestyle changes and planning to the patients well ahead.

Related Work

Choi et al. (2016a.) applied GRU model on EHR data to predict diagnosis for next visit. Similarly, DLP Yogesh et al. 2019 used Recurrent Neural along with convolution and embedding layer to predict future utilization of services. Attention models (Choi et al. 2016b., Choi et al. 2016c., Yiming et al. 2016) also have a significant impact of training and prediction performance, which especially helps when using low volume datasets. There are few other research topics, graph embeddings (Luca et al. 2019.), code to sequence embedding (Alon et al 2018.) and embedding static and dynamic information (Cristobal et al.) which are interesting and might be investigated while model building.

Cohort

We used de-identified patient claims data provided by IQVIA. IQVIA provided 5 years (2015-2019) of claims data.

Exploratory Data Analysis

Below are few exploratory analysis notes. These helped us pick a best approach for model design and build an intuition of model performance.

- Distribution of patients among the years is highly skewed with early years having a much higher number of patients (Table 1.).
- Distribution of average number of records per patient shows very strong right skew (Table 1.).
- 0.02% patients have only 1 claim record. These records cannot be used for prediction and will be filtered from the dataset.
- Only 0.3% patients have more than 100 claims and 89% of patients who have an ER claim, have their first ER claim in their first 100 claims (by 1st 100 claims, we mean, for a patient, ordering records by claim service from date in ascending order and selecting top 100 records). This helps us understand that we can treat patients with more than 100 claims as outliers and remove them from analysis. This will also reduce the dimensions for the model and make the learning faster and memory less. So, we cutoff the claim sequence for a patient for the first 100 claims or less.
- Top 5 main diseases which result in 50% of ER visits are – Essential Hypertension (10%), Abdominal pain and other digestive/abdominal signals (10%), Other specified status (10%), Tobacco related disorders (8%), Nausea and vomiting (6%) and Personal/Family history of disease (6%). These diseases make up around 50% of patients who have an ER Visit. Patients who are predicted to have an ER Visit by our model, have a higher chance of having one of the above diseases. This helped us understand, following the claim diagnosis code sequences for a patient will be an appropriate feature for ER prediction.

		# Records			
		Per Patient			
Claim Year	# Patients	Median	Mean	Total	% ER Visit
2015	18,927	20	39	736,051	1.33%
2016	21,483	19	39	842,013	1.34%
2017	15,190	16	30	467,478	1.19%
2018	6,445	19	36	232,699	1.00%
2019	4,884	18	33	159,813	1.15%
			# Records	2,438,054	30,744

Table 1. Descriptive Metrics of overall dataset

- In the dataset we have when we compare the age of the patient vs. ER visit, we found that younger/mid age patients tend to visit ER visit frequently than the older patients. This suggest that there is a strong correlation between the age as a feature with the target ER visit.
- When we compare the gender vs. ER visit, we found that the female patients end up in ER visit frequently than the male patients.

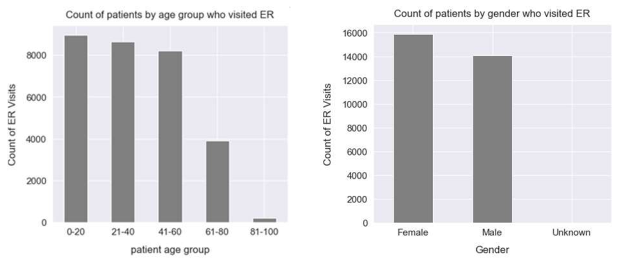


Figure 1 (left –Age distribution, right – Gender distribution)

Data Challenges

Missing data and low volume of cleaned data were a challenge during this experiment:

- Data availability - Neural networks potentially need a lot of data (i.e., big data) to give good performance. Since we do not have access to such a huge set of data, we plan to prove conceptually that our model works and would improve when it is trained with more data.
- 2% of the records have a missing record type which does not have revenue code information. Record Type is a very important feature because it helps in identifying the type of claim (such Facility, Pharmacy, Surgery, Ancillary etc.,) and critical for target labelling (ER vs. not ER).
- 29% of claims are Pharmacy claims (record type = P), since pharmacy claims do not have diagnostic codes, these were removed. This ultimately reduced our dataset by another 30%.

Based on the above listed analysis, we performed these data cleaning steps:

- Remove Pharmacy claims (i.e., records with record type value as P).
- Remove records with null/missing record type.
- Remove patients who have just one claim.
- Sort all records by patient id and claim service start date.

Identifying ER Visits from Claim Revenue Code

The claim revenue codes represent a high-level description of services performed by a hospital/other facility. Claims with the revenue code value in the below list is used to identify the ER claims from the dataset and label them as target.

Revenue Code	Revenue Description
0450 – 0459	EMERGENCY ROOM
0981	PROFESSIONAL FEES – EMERGENCY ROOM

Table 2. Revenue Code which were used for ER Identification

Mapping Diagnostic Codes to CCS Labels

The Clinical Classifications Software (CCS) is a diagnosis and procedure categorization scheme that can be employed in many types of projects analyzing data on diagnoses and procedures. We got all ICD9 & ICD10 diagnostic codes to their corresponding CCS codes mapping file from H.CUP website. These CCS codes were grouped by their high-level description of the disease. Each group was then mapped to a label. Thus, we were able to reduce more than 72,000 unique diagnostic codes in the claim file to 539 unique labels for disease classification. This is a very crucial step for the model building since training the model is simpler as the number of embeddings for diagnosis codes is significantly reduced which will help with model training and learning process.

Generating Diagnostic Code Sequences

Data was provided in columnar format indexed by a combination of patient id, claim number and service from date. To feed this into the neural network, this data was converted into list of sequences - diagnostic code in time step sequence (based on claim service from date). At the end of this step, each patient would have N diagnostic codes for each visit and M visits for each patient. If a patient does not all N diagnostic codes, then 0's is padded to corresponding places. Similarly, if a patient does not have all M visits, then N sets of 0 are padded for each visit. (Here, K – number of patients, M – maximum number of visits, N – maximum number of diagnostic codes).

Generating Static features

We created age feature using Year of Birth data from patient demographics' data. These features, age along with gender are fed as separate embedding sequences to a linear fully connected layer and finally concatenated with the diagnosis time sequence output.

Methods

Baseline Models

Random Forest

We plan to use Random Forest as a baseline to compare performance of deep learning model, as the efficacy of Random Forest classifiers is well proven when training the model with the sequentially arranged diagnosis codes. Random Forest combines the predictions of many decision trees into a single ensemble model, which is competitive enough to a dense neural network-based training model while avoiding overfitting. In our experiment, we used Random Forest with 1000 decision trees to train the model. It is also less computationally expensive for our experimentation as it does not need GPU to finish training. We used Accuracy, Precision and Recall as metrics to measure and compare the performance over the 2018-year claims data from the dataset, as these are more reliable way to compare the performance for any binary classification problem.

We used AUC as the primary metric. Table below shows the AUC of the proposed approaches in comparison with baselines on the selected 2018-year claims dataset. For each patient in the testing set, we predict whether the patient would have subsequent ER visit using his/her historical embedded diagnosis sequences. The Area Under the Curve (AUC) is the measure of the ability of a classifier to distinguish between classes and is used as a summary of the ROC curve. The higher the AUC, the better the performance of the model at distinguishing between the positive and negative classes. As you can observe, GRU performs the best among the models considered.

LSTM

We also used LSTM as another baseline which is used for sequences prediction. In this case, a simple LSTM layer and a fully connected layer, passed through a sigmoid function, is used for the ER prediction. Though LSTM can perform equally good as GRU for time series sequence prediction for our model we still used GRU since it is less complex when compared to LSTM

Claim Year	Method	AUC	Precision	Recall	f-score
2018	Random Forest	0.53	0.36	0.10	0.15
2018	LSTM	0.77	0.49	0.22	0.51

Table 3. Performance of Random Forest and LSTM Models

Model

The basic component of our model is Gated Recurrent Unit (GRU) which is good at learning long range sequences. We considered LSTM which has very similar results as GRU but GRU is much faster and has a less complicated structure. GRU trains very well on less training data as is the case in ours when compared to LSTM.

Our single classification problem was to predict the probability of the patient visiting ER and tag them as a high-risk patient if the probability is greater than 50%.

Recurrent neural network (RNN) captures the characteristics of the input sequence by recursively updating its internal state. We use the RNN only on the temporal data of the patient visits. All static information like demographics (Gender, Age), since they are not dynamic, are not fed into the recurrent neural network. But we do use this data by combining the output from GRU with demographics information to finally predict the classification.

Code Embedding layer

All patient visits are unequal in length and hence were padded to equal lengths, followed by calculation of corresponding masks for the true visits. Each visit, represented by the corresponding diagnosis codes that represent the diseases, is then converted into embeddings along the dimensions of the input dimension size.

RNN

Below is the core of the model that uses each visit of the patient in temporal order and feeds the set of diagnostic codes as embeddings and the input vector at each visit. The RNN is self-looped and is fed different diagnostic codes at each visit. The model learns to remember the important characteristics and forget the ones that does not impact the output. The learning rate is much faster in GRU with lesser data.

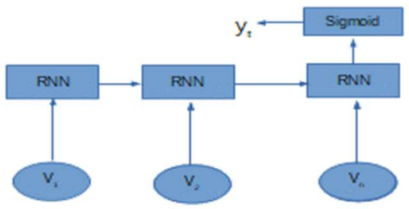


Figure 2. Generic RNN architecture

The diagnostic embeddings which represent the feature vector V1 (t=1) is fed into RNN as first step. At t=2, the output from the first hidden state is then fed into the same RNN with both output and the next visit diagnostic codes embeddings. This process is repeated 128 times (number of hidden states). We apply mask on the output from last hidden state to find the true last hidden state of the last visit.

For activation function, we used a sigmoid at each hidden state to get the output and feed it to the next hidden state. The last activation function when applied on the last hidden state would give us the desired y_t.

A GRU has two gates, namely a reset gate and an update gate z. The reset gate determines the combination of the new input vector, which are the embeddings that represent diagnostic codes for that visit, and the previous memory, which allows the hidden layer to drop irrelevant information that is not useful to the prediction. The update gate controls how much information from the previous hidden layer should be kept around. The mathematical formulation of GRU can be described as follows:

$$z = \sigma(W_z \cdot x_t + U_z \cdot h_{(t-1)} + b_z)$$
$$r = \sigma(W_r \cdot x_t + U_r \cdot h_{(t-1)} + b_r)$$
$$\tilde{h} = tanh(W_h \cdot x_t + r * U_h \cdot h_{(t-1)} + b_z)$$
$$h = z * h_{(t-1)} + (1 - z) * \tilde{h}$$

where · denotes the entry-wise product, σ is the activation function, r and z represent the reset gate and update gate at time t respectively, $\tilde{h}$ is the intermediate memory unit, and $h_{(t-1)}$ is the hidden unit. Matrices W_r, W_z, U_r, U_z along with vectors b_r and b_z are model parameters to be learned. At time t, we take the hidden state h_t to predict the labels of time $(t + 1)$.

Attention Mechanism

As mentioned in the above section, RNN can remember the past information for future prediction. However, it is limited to only a few latest steps, with more impact from later ones, and may not be able to discover major influences from earlier timestamps. Therefore, we apply attention mechanisms to memorize the effect from long-time dependencies.

Self-attention, also known as intra-attention, is designed to capture dependencies between the tokens belonging to the same sequence. Self-attention has been used in many NLP tasks such as machine translation and language models. Multiheaded attention consists of multiple copies of these self-attentive layers, all receiving the same input. The self-attention module takes a key-value pair (k_t, v_t) and a query q_t to compute the output o_t for each time step t. The self-attention coefficient is computed by first stacking together the q_t , k_t and v_t values along the time-axis to obtain $Q, K\ and\ V$, respectively using the following equation,

$$Attention\ (Q, K, V) = SoftMax\ (QKT / \sqrt{T}) \times V$$

T is the length of the sequence. Since we care about learning dependencies between each time step, in our case, the key, value and query are all copies of the input sequence, X. For multi-headed self-attention, these attention weights are calculated multiple times in parallel and the resulting vectors are concatenated together. Finally, these concatenated vectors are projected onto the output space using a fully connected layer. In the following equation, V_o, WQ_i, $WK_i\ and\ WV_i$ are the parameters learned by the model.

Multiheaded (Q, K, V) = Concat ($head_0, head_1, \ldots, head_h$)Wo

where $head_i$= Attention (QWQ i, KK i, V V i)

Attention coefficients is aggregated across the time-steps using max pooling.

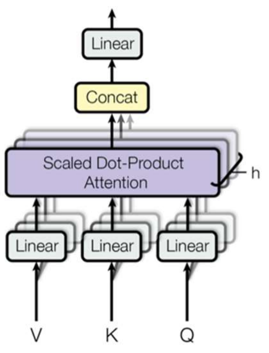


Figure 3. Attention Mechanism process illustration.

In our model, this sequence of attention coefficients is aggregated across the time-steps using max pooling before being combined with the output from the last hidden state of the GRU, using element-wise sum, and finally passed onto next layer.

The architecture of the multi headed attention model is shown below. The transformer adopts the scaled dot-product attention: the output is a weighted sum of the values, where the weight assigned to each value is determined by the dot-product of the query with all the keys.

Static Information (Demographics) Process

The output of the RNN is then combined with demographics information, i.e., age and gender and passed to a fully connected layer. We predict the probability of a patient to have an ER Visit (given a feature set consisting of diagnostic codes). The model has options to not include static information like demographics and run too. We saw the accuracy increased when we included the demographics information in addition to the temporal data of visits alone.

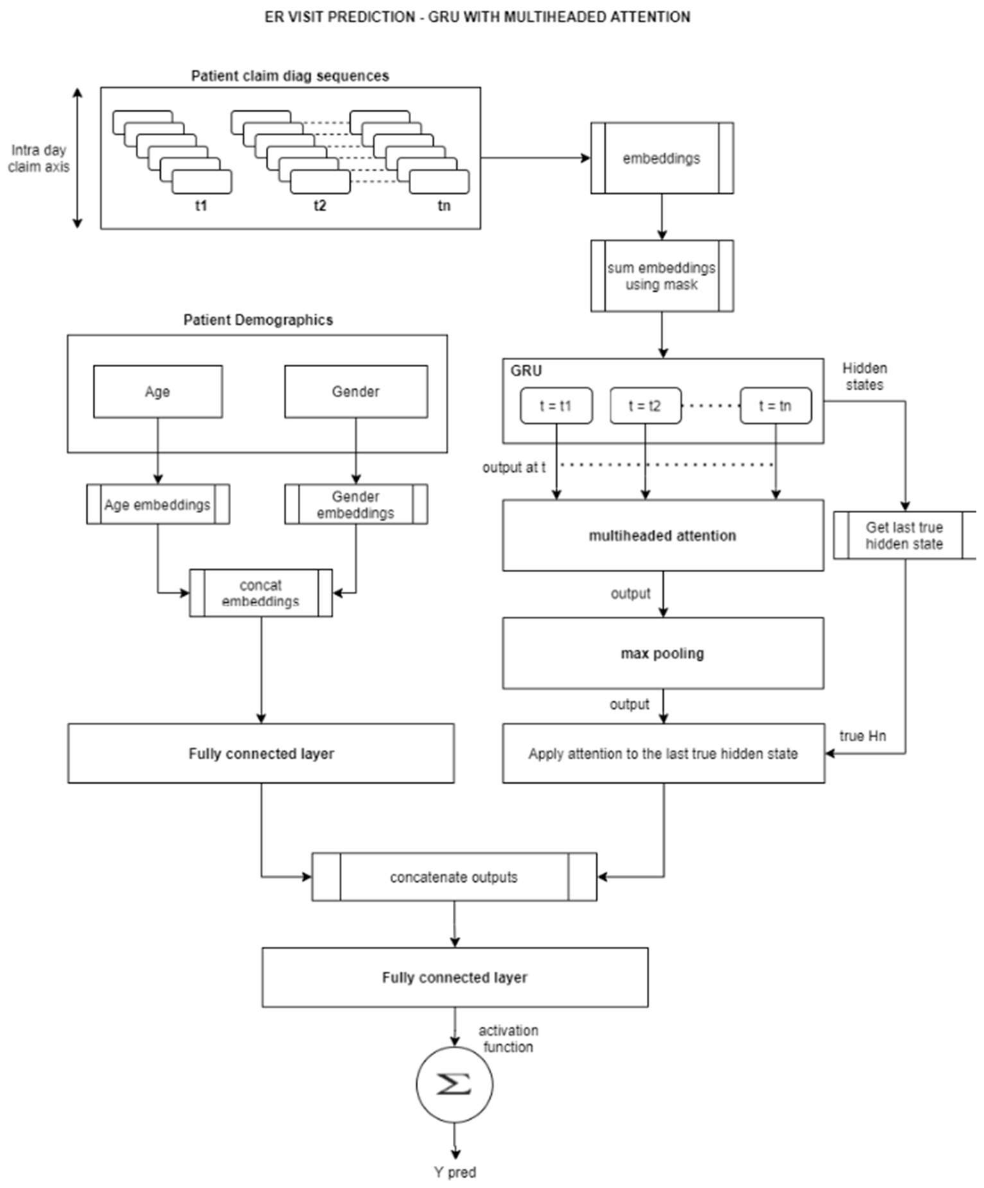


Figure 4. GRU Model with multihead attention

Loss function Binary Cross-Entropy / Log Loss

We are using Binary Cross Entropy loss function since we have a classification problem.

$$H_p(q) = -\frac{1}{N}\sum_{i=1}^{N} y_i \cdot log(p(y_i)) + (1 - y_i) \cdot log(1 - p(y_i))$$

where y is the label (1 for in ER and 0 for being in ER) and p(y) is the predicted probability of the patient being admitted into ER for all given N diagnosis codes.

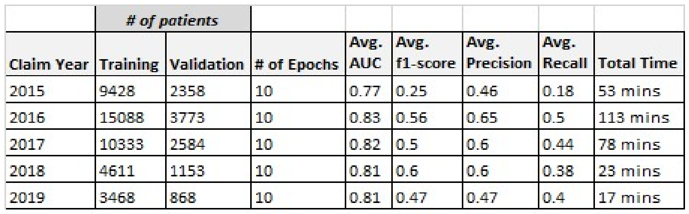
Reading this formula, it tells you that, for each patient being in ER (y=1), it adds log(p(y)) to the loss, that is, the log probability of it being tagged as ER. Conversely, it adds log(1-p(y)), that is, the log probability of it being not in ER, for each patient not being in ER (*y=0*).

Activation Function

The activation function sigmoid is run to get the final output and the probability of the person visiting ER.

Results

The below table summarizes the results for the GRU model for the different claim year periods (2015 to 2019) for 10 epochs with batch size as 32. The model is performing consistently with accuracy as metric whereas f1-score is biased. Precision (identify true positive correctly) and Recall (identify true negative correctly) is also consistent except for 2015 and 2019. As mentioned, the number of ER claims is very less in the dataset but still the model is performing well with the small dataset. As suggested before with more data to train, the model can perform much better in accuracy than random forest or other DL methods.



	# of patients							
Claim Year	Training	Validation	# of Epochs	Avg. AUC	Avg. f1-score	Avg. Precision	Avg. Recall	Total Time
2015	9428	2358	10	0.77	0.25	0.46	0.18	53 mins
2016	15088	3773	10	0.83	0.56	0.65	0.5	113 mins
2017	10333	2584	10	0.82	0.5	0.6	0.44	78 mins
2018	4611	1153	10	0.81	0.6	0.6	0.38	23 mins
2019	3468	868	10	0.81	0.47	0.47	0.4	17 mins

Table 4. Metrics of GRU Model

Shown below is the plot of training loss and AUC/f1score for the 2019 claim year data

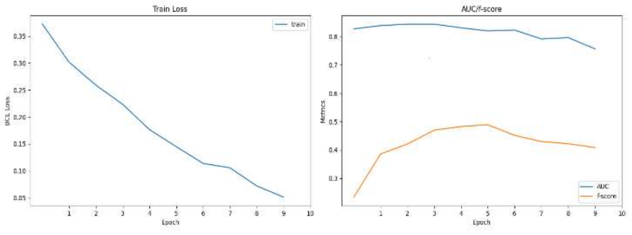


Figure 5. 2019 claim year data performance plot

The plot shows the training loss is gradually decreasing with each epoch whereas the AUC/f1score converges in 5 to 6 epochs. This suggests that the training for this model could be completed within few epochs.

Conclusion

In this paper, we introduce attention based RNN architecture to predict patients' likelihood of subsequent Emergency Room Visit. We train and learn from the diagnoses code sequences simultaneously, based on patients' historical claim records. By employing recurrent neural network, our model can remember hidden knowledge learned from previous visits. Multi headed attention mechanisms allow us to interpret the prediction results reasonably. Experimental results on IQVIA claims datasets show the effectiveness of the proposed attention based RNN models. Analysis shows that the attention mechanisms can assign meaningful weights to previous visits when predicting the future visit information. The proposed approach can be widely used for the prediction of a variety of different diseases or diagnoses in subsequent visits based on historical claims information.

We can clearly observe that the RNN based methods outperform other baselines. This owes to the capability of RNN in memorizing long-term dependencies of patient's longitudinal health records, and the attention-based mechanisms can further enhance this capability. This proposed approach can be tailored for specific use cases, such as –

- Recommending health measures to patient who show high probability of having an ER Visit.
- Pre-allocate health officials' schedules by looking at diseases in patients predicted to have an ER Visit in near future.
- Evaluate and allocate appropriate hospital departmental budgets having high propensity of patient ER visits.

Future Considerations

There are several enhancements which could be made to this model. Improvements related to data could be to include Pharmacy related records using another GRU network to explore relationship of Pharmacy code sequences. We could also impute missing data using best imputation methods and test performance improvement. We could also leverage time delta sequences (like Doctor AI) to amplify the signals. Improvement related to model architecture would be to explore graph neural network, as it takes the tree-based hierarchical relationships between diseases into account.

Contributions

Contributors: Raja Mukherjee (Raja), Balaji Sathyamurthy (Balaji), Gowri Shankar Ramanan (Gowri), Ramasamy Ramanathan (Ram)

During the project draft preparation, Raja Mukherjee and Gowri contributed to project draft documentation, while Ram and Balaji contributed to experimental evaluation. For final project report preparation, everyone gave feedback and contributed equally through several iterations. Data preprocessing and Baseline models were prepared by Gowri and Raja. Final code, including GRU and LSTM models were tested, validated and packaged by Balaji and Ram.

References

Edward Choi, Mohammad Taha Baha Dori, Andy Schuetz, Walter F Stewart, and Jimeng Sun.
Doctor ai: Predicting clinical events via recurrent neural networks. In Machine Learning for Healthcare Conference, pages 301–318, 2016a.

Yogesh Kumar, Henri Salo, Tuomo Nieminen, Kristian Vepsalainen, Sangita Kulathinal, Pekka Marttinen ; Proceedings of the Machine Learning for Health NeurIPS Workshop, PMLR 116:93-111, 2020.

Edward Choi, Mohammad Taha Bahadori, Jimeng Sun, Joshua Kulas, Andy Schuetz, and
Walter Stewart. RETAIN: An interpretable predictive model for healthcare using reverse
time attention mechanism. In Advances in Neural Information Processing Systems, pages 3504–3512, 2016b.

Edward Choi, Andy Schuetz, Walter F Stewart, and Jimeng Sun. Medical concept representation
learning from electronic health records and its application on heart failure prediction.
arXiv preprint arXiv:1602.03686, 2016c.

Yiming Cui, Zhipeng Chen, Si Wei, Shijin Wang, Ting Liu, and Guoping Hu. Attention-overattention
neural networks for reading comprehension. arXiv preprint arXiv:1607.04423,2016.

Sparkle Russell-Puleri. https://sparklerussell.com/post/gated-recurrent-units-explained-using-matrices-part-1/

Cristobal Esteban, Oliver Staeck, Stepahn Baier, Yinchong Yang, Volker Tresp. Predicting. Clinical Event by Combining Static and Dynamic Information using Recurrent Neural Networks https://arxiv.org/pdf/1602.02685.pdf

Luca Massarelli, Giuseppe A. Di Luna, Fabio Petroni, Leonardo Querzoni, Roberto Baldoni. Investigating Graph Embedding Neural Networks with Unsupervised Features Extraction for Binary Analysis. http://dx.doi.org/10.14722/bar.2019.23020

U. Alon, O. Levy, and E. Yahav, “code2seq: Generating sequences from structured representations of code,”CoRR, vol. 1808.01400, 2018.

Moore BJ (IBM Watson Health), Stocks C (AHRQ), Owens PL (AHRQ). Trends in Emergency Department Visits, 2006–2014. HCUP Statistical Brief #227. September 2017. Agency for Healthcare Research and Quality, Rockville, MD. www.hcup-us.ahrq.gov/reports/statbriefs/sb227-Emergency-Department-Visit- Trends.pdf

Suo Q, Ma F, Canino G, et al. A Multi-Task Framework for Monitoring Health Conditions via Attention-based Recurrent Neural Networks. AMIA Annu Symp Proc. 2018;2017:1665-1674. Published 2018 Apr 16.

Kyunghyun Cho et.al., Learning Phrase Representation using RNN Encode-Decoder for Statistical Machine Translation 2014 arXiv:1406.1078

R. Dey and F. M. Salem, "Gate-variants of Gated Recurrent Unit (GRU) neural networks," 2017 IEEE 60th International Midwest Symposium on Circuits and Systems (MWSCAS), 2017, pp. 1597-1600, doi: 10.1109/MWSCAS.2017.8053243.

ICD9 and ICD10 to CCS code conversion: https://www.hcup-us.ahrq.gov/toolssoftware/ccsr/DXCCSR-Reference-File-v2020-2.xlsx