2019

Opioid Misuse Detection in Hospitalized Patients Using Convolutional Neural Networks

Brihat Sharma

LOYOLA UNIVERSITY CHICAGO

OPIOID MISUSE DETECTION IN HOSPITALIZED PATIENTS USING CONVOLUTIONAL NEURAL NETWORKS

A THESIS SUBMITTED TO
THE FACULTY OF GRADUATE SCHOOL
IN CANDIDACY FOR THE DEGREE OF
MASTER OF SCIENCE

PROGRAM IN COMPUTER SCIENCE

BY
BRIHAT SHARMA
CHICAGO, IL
MAY, 2019

ACKNOWLEDGEMENTS

I would like to thank my professor, and my advisor Dr. Dmirty Dligach, who gave me this opportunity to learn, grow and gain experience in this field of work. His guidance, support, and mentorship has been valuable assets for this successful project.

I would also like to thank Dr. Majid Afshar who has been supportive throughout the project. His knowledge and understanding of the clinical domain has been helpful for me to understand the importance of this project from a healthcare background.

Many thanks to my parents who have supported me tremendously during my college years and their blessings have helped me to overcome during the time of difficulties.

And lastly, I would like to thank all my friends and colleagues who helped me throughout this journey.

For my parents.

“Last night I woke up with someone squeezing my hand. It was my other hand.”
-William S. Burroughs, Naked Lunch

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ABSTRACT

Opioid misuse is a major public health problem in the world. In 2016, 11.3 million people were reported to misuse opioids in the US only. Opioid-related inpatient and emergency department visits have increased by 64 percent and the rate of opioid-related visits has nearly doubled between 2009 and 2014. It is thus critical for healthcare systems to detect opioid misuse cases. Patients hospitalized for consequences of their opioid misuse present an opportunity for intervention but better screening and surveillance methods are needed to guide providers. The current screening methods with self-report questionnaire data are time-consuming and difficult to perform in hospitalized patients.

In this work, I explore the use of convolutional neural networks for detecting opioid misuse cases using the text of electronic health records as input. The performance of these models is compared to the performance of a more traditional logistic regression model. Different architectures of a convolutional neural network were trained and evaluated using the area under the ROC curve. A convolutional neural network performed better by producing a score of 93.4% whereas the score produced by logistic regression was 91.4% on the test data. Different advantages and disadvantages of using a convolutional neural network over the baseline logistic regression model were also discussed.

CHAPTER 1

INTRODUCTION

There is a continuum of opioid misuse that ranges from non-medical prescription to illict use. Opioid addiction has rapidly increased in recent years. Opioid-related inpatient and emergency department visits have increased by 64 percent, and the rate of Opioid-related visits has nearly doubled between 2009 and 2014. In 2015 only, 11.3 million Americans, age more than 12, have been reported to misuse opioids (Deborah Dowell 2016). Death by opioid overdose has been predicted to rise by 147 percent from 33000 in 2015 to 87000 by 2025(Qiushi Chen 2019).

There are currently no automated methods to detect opioid misuse using Electronic Health Records (EHR). Recently, there have been evidence-based treatment guidelines developed by the American Society of Addiction Medicine(ASAM), but these methods are usually slow in terms of detection, and don't work in emergency cases where quick decision making is important to treat the patients(Kyle Kampman 2015).

Significant amount of information about the patient is captured in unstructured form as clinical notes in electronic health records. These notes are a valuable source of information about the patients, which can be used to create machine learning models to predict the patient's opioid misuse status.

Natural language processing (NLP) and machine learning (ML) have recently been used in many clinical practices. NLP has an ability to extract important features from clinical notes and make decisions that could take many hours or even days for doctors. Recently, with the use of large datasets, neural networks have helped to enhance the state-of-art result produced by traditional NLP and ML models (Anthony Rios 2015). AI pioneer Geoffrey Hilton once said, “all you need is lots and lots of data and lots of information about what the right answer is, and you'll be able to train a big neural net to do what you want.”(Sharp 2017)

In this study, a logistic regression model is used for detecting opioid misuse cases as a baseline and then convolutional neural network models are used to improve on the baseline model. Convolution neural network models have been used extensively in recent years for document classification tasks. In this study different convolutional architectures are researched and studied to predict patient opioid misuse status. Different advantages and disadvantages of using such models over the baseline logistic regression model have also been investigated.

CHAPTER 2

METHODOLOGY

This section provides insights into the methods used for the study. The section is divided into different subsections for readability. In general, it discusses the methods used to create a state-of-the-art model for detecting opioid misuse and provides details on the data set, data cleaning, and different model types.

Data

The patient information is available in the form of text in the electronic health records. These notes contain unstructured data such as doctors' notes and medical test results. The notes were manually annotated, i.e. a trained professional with oversight by an addiction specialist annotated the medical chart for each patient encounter and classified the likelihood for opioid misuse in accordance with the National Institute on Drug Abuse criteria. Manually annotated data are time consuming to label and only about 10 encounters can be completed in one hour. Loyola Medical center annotators followed these guidelines in deciding which label to assign:

1. Definitely(meets NIDA guidelines)
2. Highly Probable (more than one of "probable" classifiers, diagnosis of narcotic dependence AND suspicion of misuse in the notes with no evidence)

3. Probable (history of misuse but no current documentation, OR physician mentions drug-seeking behavior, OR evidence of other substance misuse except alcohol with prescription opioid use)
4. Definitely Not (does not meet any of the NIDA guidelines nor is there any mention of misuse in the encounter)
5. Uncertain

Anything probable or above is classified as postive cases and rest are classified as negative cases (Afshar 2019). For this study, a total of 1000 manually annotated encounters were provided for the study by the Loyola Medical Center, which consists of 936 unique patients.

Data Cleaning

The provided data typically consists of a mixture of integers and strings. For this study, the data is cleaned to make them ready for training. Apache cTAKES is used for tokenization, where the alphanumeric data are separated into integers and strings, and all the integers are replaced by a string named as "number_token". The whole process took about 2 hours to complete. The tokenization is done by space between the words; e.g. *"Patient's back aches and he has been consuming 15mg of naxolane for 3month"* is changed into *"Patient s back aches and he has been consuming number_token mg of naxolane for number_token month"*.

Data Split

The cleaned data is then separated into train, validation and test sets. The split is done first into 80% and 20 % for train and test set. After that, the train set is again divided into train and validation set. The ratio of positive to negative cases is 1:2 for the entire dataset. The table 1 below shows the data split for individual subset.

Table 1: Data Split Overview

Data split into train, validation and test set			
Data	Positive	Negative	Total
Train	200	400	600
Validation	48	100	148
Test	62	126	188

Model Type

Two different models are used to detect the misuse cases. First, logistic regression is used as a baseline and then deep learning model is used to improve on the performance.

Logistic Regression Model

Logistic regression model projects the data into a euclidean space and forms a line to separate them into two classes (binary classification task). This helps to understand relationships between an independent variable and a dependent variable. Scikit-learn libary is used to implement a logistic regression model and only unigram features are used for training the model.

Deep Learning Model

A convolutional neural network (CNN) is used to enhance the overall performance of the baseline model. CNN is a type of a deep neural network, in which the units of the network consists of learnable weights and biases (Karpathy 2019). CNN contains four main layers such as convolution layer, ReLU layer, pooling layer, and a fully connected layer. Convolutional layer contains filters with different sizes, and with sharable weights it has an ability to learn feature quickly even with shallow network on the contrary feed forward neural network needs high number of neurons to learn features if the input dimension is large. The ReLU layer helps to create a non-linearity, the pooling layer reduces the dimension of the learned features either by averaging them or by taking the maximum value. The fully connected layer is a dense layer similar to the feed forward neural network consisting of neurons. In recent years, CNN has shown state-of-the-art performance on many NLP tasks (Anthony Rios 2015). For this study, CNN was implemented using a TensorFlow API known as Keras, version 2.2.4 and trained on GPU, NVIDIA 1080 TI.

Table 3 provides insights on different types of CNNs used for the experiments to train the model. In the table, CNN followed by a number refers to a key which can be mapped to the same CNN throughout this paper later in the sections. Some of the layers used in the tables are convolutional 1D layer, GlobalMaxPooling is a pooling layer which only extracts the highest weighted value from the list of values extracted by the filters of the convolutional 1D layer. Dropout masks random connections before passing the remaining information to the next layer. Dropout is used in neural networks to combat overfitting. Dense layer is a fully-connected layer, and the sigmoid layer is a dense layer with only one unit with sigmoid function in it used for binary classification.

Table 2 shows abbreviation for each layer, which will be used used in table 3

Table 2: Convolutional Neural Network Layers

Layers Abbreviations	
Layers	Abbreviations
Convolutional 1D Layer	Conv1D
GlobalMaxPooling	GMP
Dropout	DP
DenseLayer	DL

In the table 3, “-” represents sequential connection and “-/-” represents parallel connection.

Table 3: Convolution Neural Network Architecture

Convolution Neural Network with Layers Overview		
Key	Model	Layers
CNN1	Sequential	Conv1D - GMP - DP - Sigmoid
CNN2	Sequential	Conv1D - GMP - DP - DL - Sigmoid
CNN3	Sequential	Conv1D - Conv1D - GMP - DP - DL - Sigmoid
CNN4	Functional API	[(Conv1D - GMP) -/- (Conv1D - GMP) -/- (Con1D - GMP)] - DP - Sigmoid
CNN5	Functional API	[(Conv1D - GMP) -/- (Conv1D - GMP) -/- (Con1D - GMP)] - DP - DL - Sigmoid

Training Process

Pre-Processing

The training data is converted into bag of words model where uni-grams are kept along with their occurrence for each documents for logistic regression. Then term frequency - inverse document frequency transform is applied to emphasise the occurrence of highly discriminative words and provide weights on it. The data is then converted into a matrix where each word is assigned a weight. Each row of this matrix represents the document and the columns represent unique n-grams.

For CNN models, the documents are converted into sequences of integers (that represent words) of a fixed length. Then an embedding layer is created which maps the integers to vectors (known as 'word embeddings'). In this study, the embedding layer was initialized randomly and learned during the training.

The models are trained on the training data set and tuned on the validation set. In this process, different sets of hyper parameters are used to get the best result on the validation set. More traditional 10-cross fold validation is not used for logistic regression and CNN because finding the hyper-parameter is already time consuming for CNN models and they are rarely used in research for such experiments. After the hyper-parameters were found, the best model is used to label the instances in the test set.

Hyper-parameter Tuning

Hyper-parameter tuning is one of the most important parts of the model training process. In order to get the best model on the training set, one has to keep changing the hyper-parameters until best model is found. In logistic regression model experiments a technique called grid search is used which checks each pre-defined hyper-parameter combination using the validation set and reports the best result. In the table 4 different C value are used, which is know as the inverse of regularization strength. And in term of regularization, L1 and L2 are used where L1 adds absolute values of coefficients as a penalty term and L2 adds square values of the coefficient as a penalty term (Nagpal 2019).

Table 4: Logistic Regression Model Hyper-Parameters

Logistic Regression with list of hyper-parameter space	
Model	Parameters
LR	class_weight = [balanced, unbalanced] C = [0.001, 0.01, 0.1, 1, 10, 100, 1000] penalty = [L1, L2]

For the non-linear model a grid search was not possible because training convolutional neural networks takes much longer than training logistic regression. Hence, random search was implemented. Random search operates by taking a sample from the given hyper-parameter space and trains the model. (James Bergstra 2012) Each sampled configuration is scored with respect to its performance on the validation set. The best-performing configuration is then used to train a model and evaluated using the test set. The list of hyper-parameters are shown in the next chapter.

Table 5: Convolutional Neural Network Model Hyper-Parameters

CNN with list of hyper-parameters space	
Model	Parameters
CNN	Filters = [32, 64, 128, 256, 512, 1024, 2048] FilterSize = [2, 3, 4, 5] DenseLayerUnits = [32, 64, 128, 512, 1024] LearningRate = [0.1, 0.01, 0.001, 0.0001] Optimizer = [Adam, Rmsprop, SGD, Adagrad]

CHAPTER 3

RESULT

This section discusses the results produced by different models. Tables and plots are shown to provide a detailed understanding of the model performances.

Metrics

Area under the receiver operating characteristics curve (ROC_AUC) from prediction score is used to examine the discrimination characteristics of a model. ROC_AUC score evaluates the diagnostic ability of a binary classifier as its discrimination threshold is varied. A naive model will give a range of ROC_AUC score in around 0.5.

Hyper Parameters Comparisons

Table 6 shows the best hyper-parameters for Logistic regression

Table 6: Logistic Regression Hyper-Parameters

Logistic Regression with best parameters	
Model	Parameters
LR	class_weight = balanced, C = 1000, penalty = l1

Table 7 shows best hyper-parameters for each CNN architectures. The key for the table below can be mapped to the key of table 3. Due to the size of the GPU batch_size of only 1 is used for the experiment.

Table 7: Convolution Neural Network Best Hyper-Parameters

Convolution neural network with best parameters	
Key	Parameters
CNN1	C1Filters = 512, C1FilterSize = 2, GlobalMaxPoolingLayer, Dropout = 0.5, LearningRate = 0.0001, Adam
CNN2	C1Filters = 512, C1FilterSize = 3, GlobalMaxPoolingLayer, Dropout = 0.5, DenseLayer = 1024, LearningRate = 0.001, Adam
CNN3	C1Filters = 1024, C1FilterSize = 2, C2Filters = 256, C2FilterSize = 3, GlobalMaxPoolingLayer, Dropout = 0.25, DenseLayer = 1024, LearningRate = 0.0001, Adam
CNN4	C1C2C3Filter = 1024, FilterSize = 2, 3, 4 GlobalMaxPoolingLayer, Dropout = 0.5, LearningRate = 0.0001, Adam
CNN5	C1C2C3Filter = 512, 1024, 512, FilterSize = 2, 3, 4 GlobalMaxPoolingLayer, Dropout = 0.25, LearningRate = 0.0001, dense = 512, Adam

The table 8 shows the parameter range in bold that are picked up by the random search for all the above architectures. This provides a better insight on the parameter search space.

Table 8: Hyper-Parameters Range by Random Search

CNN with picked hyper-parameters space by random search	
Model	Parameters
CNN	Filters = [32, 64, 128, **256**, **512**, **1024**, 2048] FilterSize = [**2**, **3**, **4**, 5] DenseLayerUnits = [32, 64, 128, **512**, **1024**] LearningRate = [0.1, 0.01, **0.001**, **0.0001**] Optimizer = [**Adam**, Rmsprop, SGD, Adagrad]

Result Comparison

In the tables below, ROC_AUC, Positive Predictive Value(PPV), Sensitivity, F1 scores, Negative Predictive Value(NPV) and Specificity are all shown for different models. The tables are separated between Logistic Regression and Neural Networks.

Table 9: Validation Score for Logistic Regression Model

Logistic Regression(LR) Score							
Model	ROC_AUC	PPV	Sensitivity	F1_P	NPV	Specificity	F1_N
LR	0.865	0.81	0.73	0.77	0.88	0.92	0.9

The table 10 below shows the results for convolutional neural network models. The models can be mapped to the key from table 3 in the Model chapter.

Table 10: Validation Performance for the Convolutional Neural Network Models

Convolution Neural Network(CNN) Validation Score							
Model	ROC_AUC	PPV	Sensitivity	F1_P	NPV	Specificity	F1_N
CNN1	0.934	0.89	0.67	0.76	0.86	0.96	0.91
CNN2	0.944	0.92	0.73	0.81	0.88	0.97	0.92
CNN3	0.941	0.94	0.71	0.81	0.88	0.98	0.92
CNN4	0.933	0.94	0.67	0.78	0.86	0.98	0.92
CNN5	0.932	0.85	0.73	0.79	0.88	0.94	0.91

The table 11 below shows the result for convolutions model. The models can be mapped to the key from table 3 in the Model chapter. The table also show the 95% binomial confidence interval on sensitivity and specificity.

Table 11: Test Performance for the best models

Test Score using the best models							
Model	ROC_AUC	PPV	Sensitivity	F1_P	NPV	Specificity	F1_N
CNN2	0.934	0.86	0.82(0.70-0.90)	0.84	0.91	0.94(0.87-0.97)	0.93
LR	0.914	0.82	0.85(0.74-0.93)	0.83	0.93	0.79(0.71-0.86)	0.92

Plot

Two different plots are shown below. The first plot 1 shows the ROC_AUC curve of different CNN models along with logistic regression. Second plot 2 only shows ROC_AUC curve for the test dataset for both the best CNN and logistic regression models.

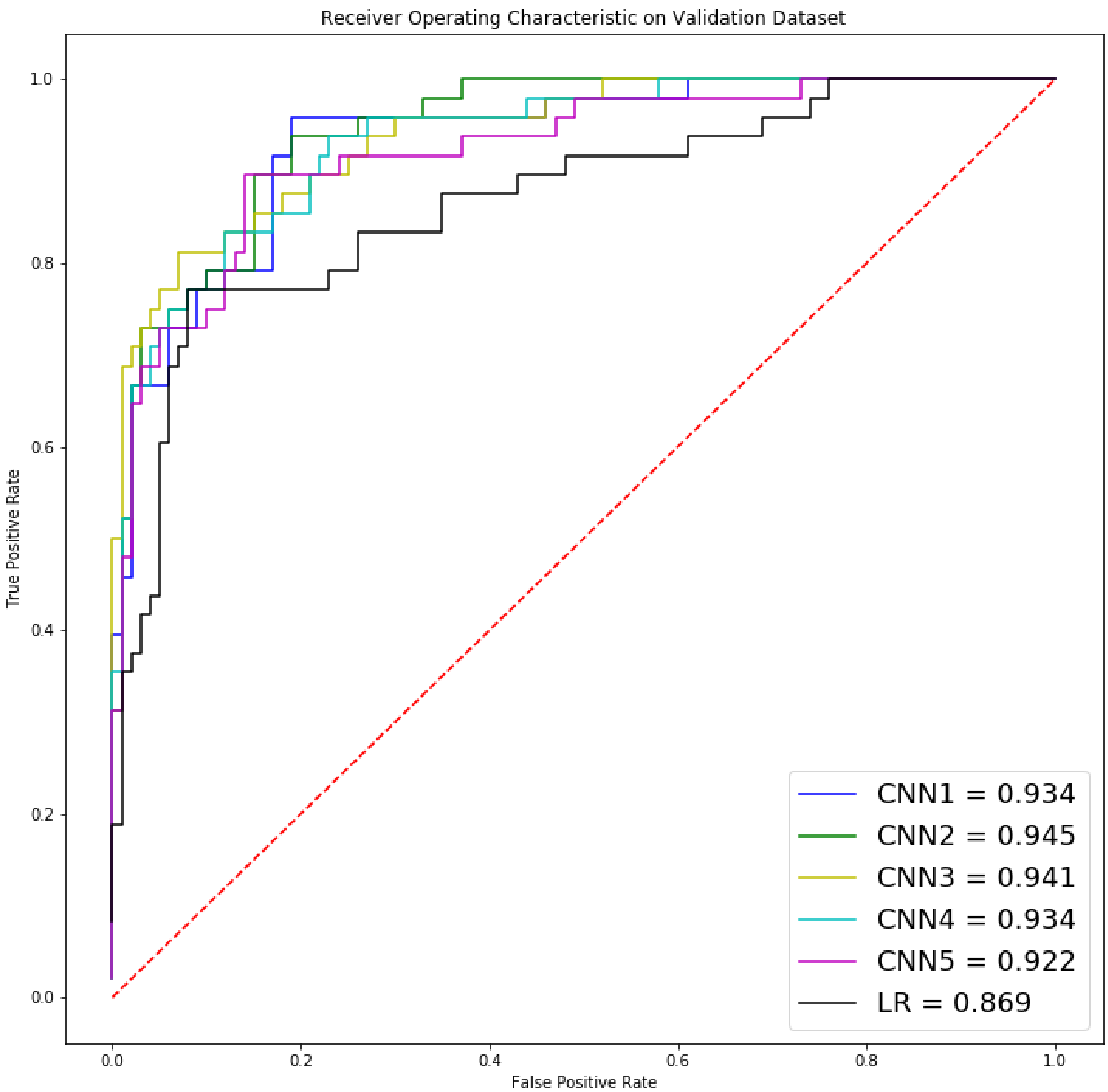

Figure 1: ROC curve for the validation data

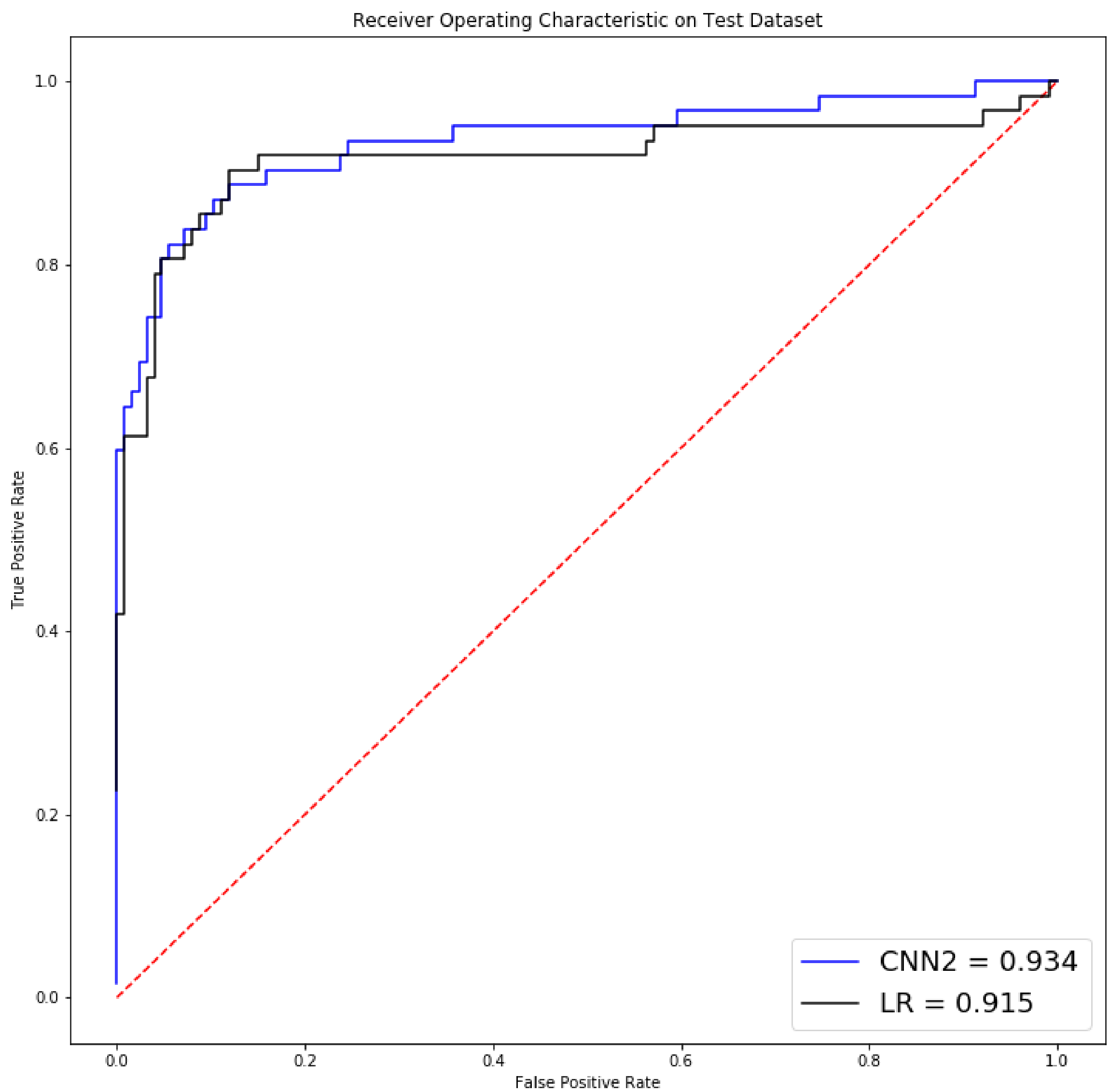

Figure 2: ROC curve for the test data

CHAPTER 4

DISCUSSION

This section discusses the results and their detailed interpretation, including such factors as time, cost and complexity while building the models.

Performance

Validation Dataset

In terms of the performance on the validation data, the best convolutional neural network outperforms logistic regression by 8 points. The best ROC_AUC score produced by logistic regression is 86.5% whereas the best convolution neural network CNN2 produces 94.4%. Increase in ROC_AUC also increases other metrics such as precision, recall, sensitivity and specificity.

Test Dataset

The result on the test set is higher for the logistic regression model with ROC_AUC score of 91.5% but for the convolution neural network it is 93.4%. The difference in performance between the validation and the test sets for logistic regression is not clear; we will investigate it further by re-training the model, with a 10-fold cross validation method and increasing the range of C values.

Overall, with this number of data a convolutional model has started to perform better than a logistic regression model but it's not immediately clear given the size of the binomial confidence interval of CNN and LR. There is a large overlap between sensitivity for LR and CNN, which can be seen from the table 11. With larger data the chances are that CNN will outperform logistic regression. The difference in performance is also likely due to the fact that neural network models in NLP operate on word embeddings, while logistic regression models work with n-gram features. While word n-grams treat words as atomic units with no notion of similarity between them, word embeddings project words into a continuous space where the degree of similarity is easily captured. Convolutional neural networks are also more flexible in terms of the features used for classification: a logistic regression model needs a pre-defined set of features, while a convolutional neural network learns the set of features it needs for classification directly from the data.

Time and Cost

In terms of cost, a linear model can be easily trained in any good performance CPU but for training a neural network, GPU is needed. For this study, two Nvidia 1080TI GPU are used, which took about 10 mins to complete training on the train data set. In order to find the best hyper-parameters, the model needs to visit each training example at least several times, hence overall time to complete the training processes becomes many hours or even days but after it's trained the time for prediction is only few seconds. If time is an important factor for training, then neural networks are not the best approach for NLP tasks. Another factor to consider is the GPU cost.

Model Complexity

Neural networks are highly expressive machine learning models but this expressiveness comes at a cost: there is typically a very large number of model parameters that need to be learned. Additionally, a large space of model hyper-parameters often has to be explored to obtain adequate performance. On the other hand, logistic regression can be trained quickly because the space of the hyper-parameters is much smaller. Finally, neural network models are not easily interpretable, which is another reason to prefer linear models.

Summary

Opioid misuse has been a major health problem in the world and at the moment we don't have quick way to detect misuse cases in hospitalized patients. In this study, a convolutional neural network and logistic regression models are used to create a model that can predict misuse status for such patients using the text of electronic medical records as an input. The convolutional neural network works better than logistic regression in terms of ROC_AUC score but more research is needed given the small size of our dataset and a large range of binomial confidence intervals for sensitivity and specificity.

REFERENCE LIST

Afshar, Majid. 2019. "Conversation in Slack Messenger." Accessed March 28.

Anthony Rios, Ramakanth Kavuluru. 2015. "Convolutional neural networks for biomedical text classification." In *Proceedings of the 6th ACM Conference on Bioinformatics, Computational Biology and Health Informatics - BCB 15.* doi:10.1145/2808719.2808746.

Deborah Dowell, Roger Chou, Tamara M. Haegerich. 2016. "National Practice Guideline for the Use of Medications in the Treatment of Addiction Involving Opioid Use." *The Journal of the American Medical Association* 315 (15). doi:10.1001/jama.2016.1464.

James Bergstra, Yoshua Bengio. 2012. "Random Search for Hyper-Parameter Optimization." *Journal of Machine Learning Research:* 281–305.

Karpathy, Andrej. 2019. "CS231n Convolutional Neural Networks for Visual Recognition." Accessed March 21. http://cs231n.github.io/convolutional-networks/.

Kyle Kampman, Margaret Jarvis. 2015. "National Practice Guideline for the Use of Medications in the Treatment of Addiction Involving Opioid Use." *Journal of Addiction Medicine* 9 (5): 358–367. doi:10.1097/adm.0000000000000166.

Nagpal, Anuja. 2019. "L1 and L2 Regularization Methods." Accessed March 21. https://towardsdatascience.com/l1-and-l2-regularization-methods-ce25e7fc831c.

Qiushi Chen, Davis T. Weaver, Marc R. Larochelle. 2019. "Prevention of Prescription Opioid Misuse and Projected Overdose Deaths in the United States." *JAMA Network Open* 2. doi:10.1001/jamanetworkopen.2018.7621.

Sharp, Alastair. 2017. "Google's Hinton says AI poses new challenges to regulators." Accessed March 21, 2019. https://www.reuters.com/article/us-alphabet-hinton/googles-hinton-says-ai-poses-new-challenges-to-regulators-idUSKBN1DY2KH.

VITA

Born and raised in Kathmandu valley of Nepal, Brihat Sharma was always been interested in understanding the nature of reality. Reading books such as 'The Elegant Universe' and wondering about the universe quenched his curiosity towards the universe since childhood. With a desire to learn more about theories of the universe, in 2010 he started his undergraduate in Physics at Ohio Wesleyan University, Delaware Ohio. His interest grew and inclined more towards rocket science and AI.

He joined Loyola University Chicago in 2017 for a graduate degree in computer science. During his masters, he took up a course offered in Machine Learning with Dr. Dmitriy Dligach. With more interest in this field, he exposed himself to Natural Language Processing projects which changed his graph of knowledge from traditional machine learning and natural language processing(NLP) to much-advanced machine learning fields such as deep learning. He was awarded the Research Assistantship(RA) position at Loyola Medical Center where his scope of work involved with larger NLP projects. Working with Dr. Afshar and Dr. Dligach, he has gained profound knowledge and experience in the medical domain along with machine learning.

During his leisure time, Sharma keeps himself busy by working on his own projects, tutoring students in machine learning and helping kids with math and physics.