Epsilon: General Machine Learning, NLP

By Yonatan Feleke <yfeleke@stanford.edu>, Ashok Poothiyot <apoothiy@stanford.edu> and Gurkanwal Brar <gbrar@vmware.com>

1 Abstract

The corporate world deals with task management in a variety of ways with each having some form of triaging process to correctly assign tickets to developers. Automation of this task has proven elusive with less than 60% accuracy of latest ML solutions. The paper explores this area and compares data from different strategies.

2 Introduction

Web and SaaS companies handle high volumes of tickets in the form of exceptions, support requests, user-reported bugs, and crash reports. Effective automation is essential to improve productivity and obviate the tedious work of manually triaging tickets. JIRA and Asana are the most widely used task and ticketing systems to tame this beast with dedicated teams that work on aggregating, triaging and assigning these tickets to the right individual or team.

Our project aims to eliminate this overhead by experimenting with supervised-learning classifiers to assign tickets to a developer. We aim to deliver higher accuracy for predicting assignee based on past tickets. In the future, we see abundant applications: automatically setting priority, EHR classification with modified featurization, automated customer support, and prioritized exception alerts.

3 Related Work

Classification on open bug reports with supervised learning is fairly common are of work, common strategies are Naive Bayes and SVM classification on a multinomial event model input featurized as bag of words. This paper replicates the mentioned experiments to set a baseline for our dataset.

A highly relevant research close to our deep neural network experiment is: “DeepTriage: Exploring the Effectiveness of Deep Learning for Bug Triaging; S Mani, A Sankaran, R Aralikatte”.

4 Dataset and Features

4.1 Dataset

The research utilizes generated http://jumble.expium.com/ with a future work plan to applying solution to Linkedin’s Foundation support tickets. The exact dataset in use can be found in our repo Jumble-for-JIRA.json

4.2 Features

The experiment implements a multinomial event model with a bag of words featurization pattern. Words in the description, body and comment sections are concatenated and converted to a feature vector with a dictionary of words that occur more than five times.

5 Methods

The project aims to evaluate different supervised learning algorithms using all previously assigned developers as the number of classes for classification. The problem is treated as a text classification problem with a novel experiment in using a deep neural network grid search to find optimal architecture and activation function.

5.1 Naive Bayes and SVM, ?Multi-class logistic regression

The section aims to evaluate results on using the currently assigned developer as a label and body of text as the feature vector to evaluate cross-entropy loss on the data set. The algorithms will be evaluated with dimensions of cross-entropy loss when tested with **lemmatization, stemming and word embeddings.**

5.2 Deep Neural network Classifier

Neural networks promise to fit highly non-linear data sets and our use case is expected to benefit highly from this property due to its freeform nature. The experiment will focus on testing with 3,5,8 layers, 8,16,32 neurons, and relu and tanh activator functions. A 5 fold experiment is used to determine loss and find the optimal architecture. The architecture is then compared to the other solutions to explore performance.

- Leverage NLP methods to form better matching among tickets for classification.

5.3 Goals for Methods

We aim to accomplish the following experiments:

1) Observe impact of **lemmatization, stemming and word embeddings on common models**
2) Experiment to find an optimal neural network architecture on 5-fold cross entropy loss
3) Compare accuracy between classical classification methods and neural networks.

Stretch Goals:

The experiments below are marked for further exploration if we are able to complete our goals above.

1)
2)

6 Experiments

Run the following experiments based on the http://jumble.expium.com/ generated datasets. A typical JIRA ticket has text body values in summary, description, comments and body sections. The featurization step concatenates all text elements and (unless otherwise specified like 6.1) builds a multinomial event model with counts tracked for words that appear more than 5 times in a bag of words model assuming words are independently selected.

6.1 NLP Uplift

We experiment with changing assumptions about our input text to see if better results can be captures.

Algorithm	Bag of words	With Lemmatization	With Stemming	Word embeddings
Naive bayes				
SVMs				
Multi-class logistic				

The Observed uplift on prediction accuracy when introducing concepts from NLP like word embedding

6.2 DNNs Architecture selection

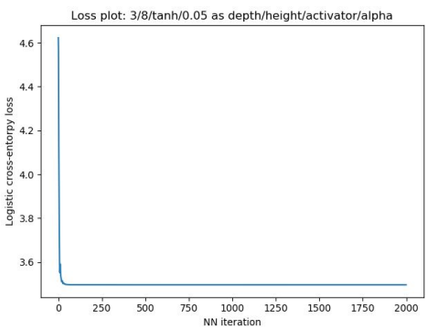
Selecting the correct DNN marks the first of experiments attempted to with the goal of improving predictions, neural networks are able to fit highly non-linear models and we hoped to see high accuracy measures. We explored cross-entropy training error across different parameters of a neural network. The following test parameters were checked.

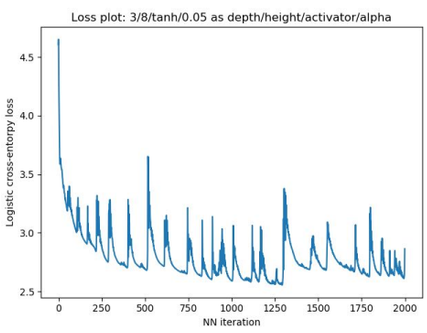
Parameter	Tested values
Activator	Reul, tanh
Learning rate (alpha)	0.5, 0.05
Backprop Iterations	1000, **2000**
Depth	3,5,8
Height	8, 16, 31

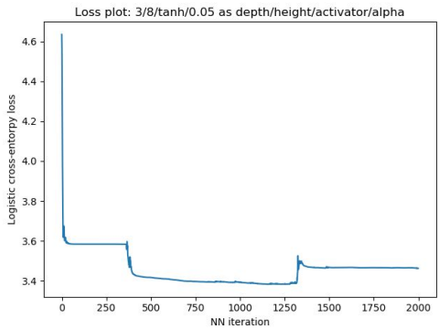
The focus on this execution is to observe the best minimization strategies during train time that would have give lowest final logistic cross-entropy loss. The experiment was run with 2,000 iterations at first but later truncated to 1,000 iterations of backpropagation. The results were fairly surprising with mix of performance with the algorithm suffering from local optima. The sklearn validation scores are listed below.

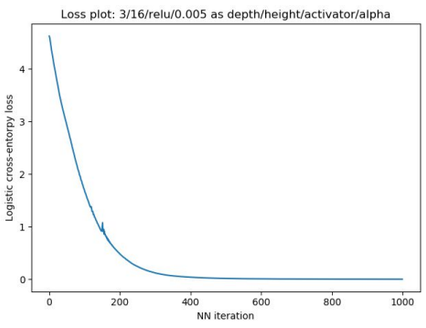
Activator	Depth	Height	Learning rate,	mean_train_score	mean_test_score
tanh	3	8	0.05	41305868585	18570222
tanh	3	8	0.005	53489560046	12377312107
tanh	5	8	0.05	103088781371	4343107167
tanh	3	16	0.05	106145714330	10400282264
tanh	3	16	0.05	15016859879	20384044969
tanh	5	32	0.05	49515867412	832679976
Relu	3	8	0.05	3882608432	15391186406
Relu	3	16	0.05	39952912058	11437985231
Relu	3	16	0.005	12514502026	1292181572
Relu	5	16	0.05	29035785681	5967770285

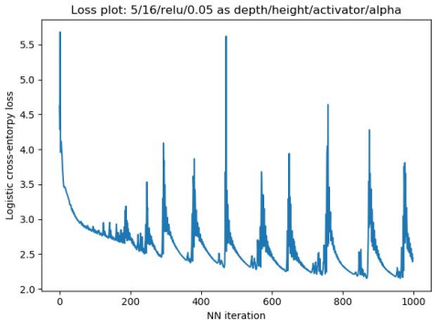
The more interesting observations were on the output of graphs compiled with varying the different parameters and looking at the train loss per-iteration. Depending on the parameters, we see wildly varying patterns that need to be tuned for better accuracy and efficient computation and at times. **It was surprising to see that taller and wider nets did not necessarily yield lower training loss.**

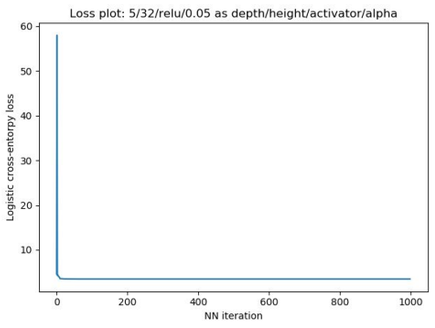












-
-

Experiments to provide answers to the following questions are interesting areas for further exploration.

- The uplift of prediction accuracy when introducing concepts from NLP like word embedding, when executed against simulated data set and LinkedIn dataset
- Would simple multi-class logistic regression beat naïve Bayes prediction performance?
- Softmax logistic regression to make multi-class classification with probabilities
- Deep learning to see what it can learning, Deep learning to learn the features needed to decide on important features for making decisions

6.3 Comparison of results

Identify a good prediction performance metrics and significance of sklearn cross validation scores.
> Capture loss trends for Naive Bayes classification and compare with NN
?> Build train, cross-validate and test infrastructure across multiple datasets.

7 Conclusions

The low accuracy rates in predicting a developer make the currently tested experiments not viable for the primary triaging
Notes on: Implement NLP concepts, advanced features and vectorization strategies.

8 Future work

8.1 Advanced Features

The current bag of words multinomial event model

- Unsupervised clustering algorithm to see correlation insights about the underlying structure of our data.
    - Identify a cluster of pathological samples and mislabelled highly correlated sets

3) Investigate more advanced features like creation dates, history and team data.
4) Methods for informing model with handcrafted rules to assist in triaging the data
5) Semi-supervised learning for samples that include an Unassigned assignee
6) Extracting exceptions and unique words like product names or error types to create additional features.
7)