proposal.md

Machine Learning Engineer Nanodegree

Capstone Proposal

Jayaram Prabhu Durairaj
August 20, 2017

Proposal

Domain Background

I have been taking Machine Learning and NLP classes online to master Natural Language Processing models. I am following cs224n and oxford-cs-deepnlp-2017 along with MLND. This project will be a comprehensive study on important NLP classification methods. Since the inception of word vectors concept, there has been a big shift from using count based models to word vector based models. However, there is not a significant difference in results because of algorithms in general [1]. There has not been a general model that could achieve state of the art results for all the tasks. In this project, I plan to try many models learned over multiple Machine Learning and NLP courses, and models suggested in many research papers which will be cited in the reference section, and some new models I will be creating based on intuition and experience.

Deep learning in medicine has been applied in a variety of applications such as image-based assessments of traumatic brain injuries, identifying diseases from ordinary radiology image data, visualizing and quantifying heart flow in the body using any MRI machine, as well as analyzing medical images to identify tumors, nearly invisible fractures, and other medical conditions.

A lot has been said during the past several years about how precision medicine and, more concretely, genetic testing is going to disrupt the way diseases like cancer are treated. However, this is only partially happening due to the large amount of manual work still required.

Problem Statement

Kaggle along with Memorial Sloan Kettering Cancer Center (MSKCC) launched "Personalized Medicine: Redefining Cancer Treatment" [21] competition, accepted by the NIPS 2017 Competition Track, because they need data scientists to help take personalized medicine to its full potential. Once sequenced, a cancer tumor can have thousands of genetic mutations. But the challenge is distinguishing the mutations that contribute to tumor growth (drivers) from the neutral mutations (passengers). Currently, this interpretation of genetic mutations is being done manually. This is a very time-consuming task where a clinical pathologist has to manually review and classify every single genetic mutation based on evidence from text-based clinical literature.

For this competition MSKCC is making available an expert-annotated knowledge base where world-class researchers and oncologists have manually annotated thousands of mutations. One needs to develop Machine Learning algorithms that, using this knowledge base as a baseline, automatically classifies genetic variations.

Datasets and Inputs

In this competition, I have to develop algorithms to classify genetic mutations based on clinical evidence (text). There are nine different classes a genetic mutation can be classified on. This is not a trivial task since interpreting clinical evidence is very challenging even for human specialists. Therefore, modeling the clinical evidence (text) will be critical for the success of my approach.

Both training and test data sets are provided via two different files. One (training/test_variants) provides the information about the genetic mutations, whereas the other (training/test_text) provides the clinical evidence (text) that our human experts used

to classify the genetic mutations. Both are linked via the ID field. Therefore the genetic mutation (row) with ID=15 in the file training_variants, was classified using the clinical evidence (text) from the row with ID=15 in the file training_text. Finally, to make it more exciting, some of the test data is machine-generated to prevent hand labeling. I have to submit all the results of the classification algorithm, and the machine-generated samples will be ignored. The following provides the file descriptions.

1. training_variants - a comma separated file containing the description of the genetic mutations used for training. Fields are ID (the id of the row used to link the mutation to the clinical evidence), Gene (the gene where this genetic mutation is located), Variation (the aminoacid change for this mutations), Class (1-9 the class this genetic mutation has been classified on)

2. training_text - a double pipe (||) delimited file that contains the clinical evidence (text) used to classify genetic mutations. Fields are ID (the id of the row used to link the clinical evidence to the genetic mutation), Text (the clinical evidence used to classify the genetic mutation)

3. test_variants - a comma separated file containing the description of the genetic mutations used for training. Fields are ID (the id of the row used to link the mutation to the clinical evidence), Gene (the gene where this genetic mutation is located), Variation (the aminoacid change for this mutations)

4. test_text - a double pipe (||) delimited file that contains the clinical evidence (text) used to classify genetic mutations. Fields are ID (the id of the row used to link the clinical evidence to the genetic mutation), Text (the clinical evidence used to classify the genetic mutation)

5. submissionSample - a sample submission file in the correct format

Dataset snippets

train data frame

	ID	Gene	Variation	Class	Text
0	0	FAM58A	Truncating Mutations	1	Cyclin-dependent kinases (CDKs) regulate a var...
1	1	CBL	W802*	2	Abstract Background Non-small cell lung canc...
2	2	CBL	Q249E	2	Abstract Background Non-small cell lung canc...
3	3	CBL	N454D	3	Recent evidence has demonstrated that acquired...
4	4	CBL	L399V	4	Oncogenic mutations in the monomeric Casitas B...

test data frame

	ID	Gene	Variation	Text
0	0	ACSL4	R570S	2. This mutation resulted in a myeloproliferat...
1	1	NAGLU	P521L	Abstract The Large Tumor Suppressor 1 (LATS1)...
2	2	PAH	L333F	Vascular endothelial growth factor receptor (V...
3	3	ING1	A148D	Inflammatory myofibroblastic tumor (IMT) is a ...
4	4	TMEM216	G77A	Abstract Retinoblastoma is a pediatric retina...

There are some missing text for Text attribute on train (1 row) and test (6 rows). Total Corpus has 45721035 words with vocabulary of 228389 words. After pre-processing some statistics on the length of the text in each train and test data rows

```
train data Text attribute
count    3321.000000
mean     1485.407709
std       845.216131
min         1.000000
25%       919.000000
50%      1215.000000
75%      1887.000000
```

```
max      7199.000000

test data Text attribute
count    5668.000000
mean     1633.366090
std       484.593756
min         1.000000
25%      1337.000000
50%      1642.000000
75%      1920.250000
max      6633.000000
```

Solution Statement

I have divided the solutions to try into three categories

1. Count-based classification methods
2. Count-based classification ensemble methods
3. Deep Learning methods

Scikit-Learn has most of the classification algorithms in really good api format. After converting the documents to Term-Frequency-Inverse-Document-Frequency matrix, the following available algorithms are used to predict the multi class log loss.

1. Multinomial Naive Bayes
2. Support Vector Machine
3. Softmax Regression
4. K Nearest Neighbour
5. Gaussian Process Classifier
6. Passive Aggresive Classifier
7. Quadratic Discriminant Analysis
8. AdaBoost Classifier
9. Random Forest Classifier
10. Extreme Randomization Trees

Some ensemble packages that are going to be used and tuned are:

1. catboost
2. xgboost
3. lightgbm

Word Vectors took the NLP community by storm with its ability to be used in deep learning models. The following are some of the Deep Learning methods that will be used for text classification.

1. Semi-supervised Convolutional Neural Networks for Text Categorization via Region Embedding [2]
2. Hierarchical Attention Networks for Document Classification [3] [4]
3. Recurrent Neural Network for Text Classification with Multi-Task Learning[5]
4. A Convolutional Neural Network for Modelling Sentences, Kalchbrenner [6]
5. Convolutional Neural Networks for Sentence Classification [7]
6. Very Deep Convolutional Networks for Text Classification [8]
7. Character-level Convolutional Networks for Text Classification [9]
8. Distributed Representations of Sentences and Documents [10]
9. Medical Text Classification using Convolutional Neural Networks [11]

More research papers that will be referenced or used as influence for new models in the project are

1. Comparative Study of CNN and RNN for Natural Language Processing [12]
2. Do Convolutional Networks need to be Deep for Text Classification ? [13]
3. How Transferable are Neural Networks in NLP Applications? [14]
4. Neural Machine Translation in Linear Time [15]
5. Generative and Discriminative Text Classification with Recurrent Neural Networks [16]

6. Combination of Convolutional and Recurrent Neural Network for Sentiment Analysis of Short Texts [17]
7. Ensemble Application of Convolutional and Recurrent Neural Networks for Multi-label Text Categorization [18]
10. CNNs for Text Categorization: Shallow Word-level vs. Deep Character-level [22]

Finally, I will create new models based on intuitions and influences gathered from this comprehensive study on text classification deep learning models.

Benchmark Model

There are no benchmarks available other than kaggle leadership board ranking. I am going to use ensemble models as benchmark models for personal validation before competing on kaggle leader-board.

Evaluation Metrics

Submissions are evaluated on Multi Class Log Loss between the predicted probability and the observed target. Multi Class Log Loss is the multi-class version of the Logarithmic Loss metric. Each observation is in one class and for each observation, I will submit a predicted probability for each class. The metric is negative the log likelihood of the model that says each test observation is chosen independently from a distribution that places the submitted probability mass on the corresponding class, for each observation.

$$logloss = -\frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{M} y_{i,j}\log(p_{i,j})$$

where N is the number of observations, M is the number of class labels, log is the natural logarithm, y_{i,j} is 1 if observation i is in class j and 0 otherwise, and p_{i,j} is the predicted probability that observation i is in class j.

Both the solution file and the submission file are CSV's where each row corresponds to one observation, and each column corresponds to a class. The solution has 1's and 0's (exactly one "1" in each row), while the submission consists of predicted probabilities.

The submitted probabilities need not sum to 1, because they will be rescaled (each is divided by the sum) so that they do before evaluation.

Project Design

I plan to use Jupyter notebook for all the necessary tasks so that the code and the comments can be well documented for reference later. All the training and validation will be performed on the training data-set.

Custom text vectorizer will be used since it has to be generic with the task at hand. Medical data has lot of new scientific vocabulary that must be considered. After reviewing the textual data from training and testing data-set, they will be pre-processed. Basic data cleaning will be performed, including converting to lower case and conforming to "UTF-8" format. Custom regular expressions will be used to remove more frequent useless words such as urls, tables and figure information. All the stop-words will be removed but lemmatization will not be performed since we might risk losing medical jargon. After applying the same pre-prossessing for all the text data available, the vocabulary for the corpus will be created.

The count based models will be used to estimate the classification task with the TF-IDF matrix. Then, using the same TF-IDF matrix, the ensemble methods will be used to estimate the log loss.

For deep learning models, it is important to get meaningful word vectors that will form the base for how well the downstream models created will perform. After obtaining the corpus (train and test data-set) vocabulary, I need to obtain vectors already trained and available such as bioNLP [18] vectors and glove [19] vectors. These vectors have been tested on large corpus. More importance will be given to the bioNLP vectors since they were trained on medical corpus. All the remaining words in the corpus that are not in bioNLP vectors will be updated with glove vectors. The remaining words will be set randomly.

Kaggle allows external textual data to be used. More data will be trained using the skip-gram [20] model to updated the remaining word vectors for capturing semantically meaningful relations. Any allowed external textual data will be pre-processed similar to training and testing data-sets. These text will be used to update the word vectors with the above created

algorithm that will capture more domain related information. This process will make the downstream models more reliable and robust.

After word-vectors are created, I will be able to implement all the above discussed models that will try to reduce log-loss and compete with other highly precise models in the competition.

Reference

[1] Improving Distributional Similarity with Lessons Learned from Word Embeddings, Omer Levy, Yoav Goldberg, Ido Dagan

[2] Semi-supervised Convolutional Neural Networks for Text Categorization via Region Embedding

[3] Hierarchical Attention Networks for Document Classification

[4] Reference blog for Hierarchical Attention Networks

[5] Recurrent Neural Network for Text Classification with Multi-Task Learning

[6] A Convolutional Neural Network for Modelling Sentences, Kalchbrenner et al. ACL 2014

[7] Convolutional Neural Networks for Sentence Classification

[8] Very Deep Convolutional Networks for Text Classification

[9] Character-level Convolutional Networks for Text Classification

[10] Distributed Representations of Sentences and Documents

[11] Medical Text Classification using Convolutional Neural Networks

[12] Comparative Study of CNN and RNN for Natural Language Processing

[13] Do Convolutional Networks need to be Deep for Text Classification ?

[14] How Transferable are Neural Networks in NLP Applications?

[15] Neural Machine Translation in Linear Time

[16] Generative and Discriminative Text Classification with Recurrent Neural Networks Combination of Convolutional and Recurrent Neural Network for Sentiment Analysis of Short Texts

[17] Ensemble Application of Convolutional and Recurrent Neural Networks for Multi-label Text Categorization

[18] Word vectors trained on PubMed and PMC texts

[19] glove vectors

[20] Efficient Estimation of Word Representations in Vector Space

[21] Personalized Medicine: Redefining Cancer Treatment

[22] Convolutional Neural Networks for Text Categorization: Shallow Word-level vs. Deep Character-level