Sentiment Analysis Report

Sarcasm Detection with Logistic Regression and Neural Networks

Data Science and Economics

Nicholas Carp

Department of Economics and Quantitative Methods
Università degli Studi di Milano
04-10-2021

Contents

1. Introduction

Sarcasm Detection is another important task that machines can do nowadays, the ability to classify sarcasm from different types of text (comments on social medias, mails, messages) is very useful, but also a complex problem. The aim of this project is to classify comments posted on reddit, more specifically to predict the fact that a parent comment will have a sarcastic comment associated to it. The methods for achieving our objective are logistic regression and neural networks.

2. Dataset and Preprocessing

2.1 Dataset

The dataset is called Sarcasm on Reddit and you can find it on Kaggle at this LINK. The dataset is quite big and is composed like it follows: The **label** that states if a comment is sarcastic or not, the **comment**, the **parent comment**, the **author**, the **subreddit** where the comment belongs to, the **score**, **ups**, **downs**, the **date** and **hour** of the comment.

Example of dataset composition

	label	comment	author	subreddit	score	ups	downs	date	created_utc	parent_comment
0	0	NC and NH.	Trumpbart	politics	2	-1	-1	2016-10	2016-10-16 23:55:23	Yeah, I get that argument. At this point, I'd ...
1	0	You do know west teams play against west teams...	Shbshb906	nba	-4	-1	-1	2016-11	2016-11-01 00:24:10	The blazers and Mavericks (The wests 5 and 6 s...
2	0	They were underdogs earlier today, but since G...	Creepeth	nfl	3	3	0	2016-09	2016-09-22 21:45:37	They're favored to win.
3	0	This meme isn't funny none of the "new york ni...	icebrotha	BlackPeopleTwitter	-8	-1	-1	2016-10	2016-10-18 21:03:47	deadass don't kill my buzz
4	0	I could use one of those tools.	cush2push	MaddenUltimateTeam	6	-1	-1	2016-12	2016-12-30 17:00:13	Yep can confirm I saw the tool they use for th...

2.2 Preprocessing Phase

In this section, we shall explain how we have preprocessed the data, in first place we have removed extra spaces and made our strings lowercase, then added spaces before and after punctuation to make words separate and replaced more than two continuous spaces with one space.

Dataset composition after preprocessing

	label	comment	author	subreddit	score	ups	downs	date	created_utc	parent_comment
0	0	nc and nh.	Trumpbart	politics	2	-1	-1	2016-10	2016-10-16 23:55:23	yeah , i get that argument. at this point , i ...
1	0	you do know west teams play against west teams...	Shbshb906	nba	-4	-1	-1	2016-11	2016-11-01 00:24:10	the blazers and mavericks (the wests 5 and 6 ...
2	0	they were underdogs earlier today , but since ...	Creepeth	nfl	3	3	0	2016-09	2016-09-22 21:45:37	they ' re favored to win.
3	0	this meme isn ' t funny none of the " new york...	icebrotha	BlackPeopleTwitter	-8	-1	-1	2016-10	2016-10-18 21:03:47	deadass don ' t kill my buzz
4	0	i could use one of those tools.	cush2push	MaddenUltimateTeam	6	-1	-1	2016-12	2016-12-30 17:00:13	yep can confirm i saw the tool they use for th...

3. Research question and methodology

The goals of the project is to find if is possible to predict a comment label based on what is the content of the parent comment. The methodology used is, after preprocessing the dataset, splitting it in Train, Validation and Test and applying logistic regression in alternative to many types of neural networks, precisely: CNN, LSTM and GRU. For the logistic regression we implemented two pipelines, one without considering the subreddit category, and the other one was taking in account also the topic of the comments. In the Neural Networks part we used Keras library for embedding, training and testing.

3.1 Logistic Regression

Logistic Regression is a Machine Learning algorithm which is also used for classification problems, it is a predictive algorithm, based on the concept of probability.

Logistic Regression uses a more complex loss function than linear one, in order to map predicted values to probabilities, we use the Sigmoid function. The function maps any real value into another value between 0 and 1. The logit equation is: $\sigma(Z) = \sigma(\beta 0 + \beta 1 * X)$

3.2 Neural Networks

An Artificial Neural network is based on a collection of connected nodes called neurons that are aggregated into layers. Each connection, called edges, can transmit a signal to other nodes, this signal is a real number, and the output of each node is computed by a function of the sum of its inputs. Mathematically NNs are obtained through the combination of simple predictors of the form: g(x) = $\sigma(w^t \times x)$.

The function is often nonlinear and is called activation function. Neurons and edges typically have a weight that adjusts as learning proceeds. The weight increases or decreases the strength of the signal at a connection. A network structure is a directed acyclic graph G = (V,E) where each node j computes a function g(v) whose argument v is the output of the nodes i such that i and j are part of E. Signal travel from input layer , to n hidden layers and finally to the output one, different layers may perform different transformations on their inputs.

Convolutional Neural Network

A Convolutional Neural Network is a special kind of multi layer neural network which can take in an input, assign learnable weights and biases to various aspects of the text and be able to differentiate one from the other. This type of NN are useful thanks to the mathematical operation called convolution that performs feature extraction at different levels. There are four main layers in simple networks: A Convolutional Layer, a Pooling Layer, a Flattening Layer and a Fully Connected Layer.

The element involved in carrying out the convolution operation in the first part of a Convolutional Layer is called the Kernel/Filter and mathematically is a matrix. The filter moves to the right with a certain Stride value till it parses the complete width. Moving on, it hops down to the beginning of the image with the same Stride Value and repeats the process until the entire image is traversed.

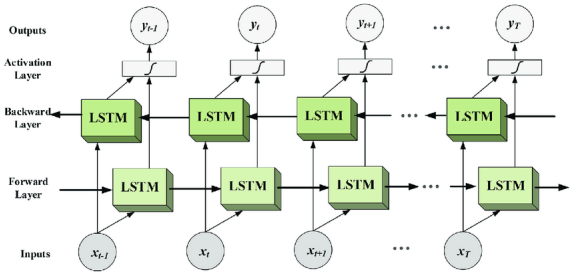
The convolution output is then passed through the activation function called ReLU (Rectified Linear Unit)that is the most commonly used activation function in deep learning models. This unit converts the data into its non-linear form, the function returns 0 if it receives any negative input, but for any positive value x it returns that value back.

The output of the convolutional layer represents the high-level features, flattened output is fed to a feed-forward neural network and backpropagation applied to every iteration of training. Over a series of epochs, the model is able to distinguish between features and classify them using the a Sigmoid function.

Long Short Term Memory

Long Short-Term Memory networks are a type of recurrent neural network capable of learning order dependence in sequence prediction problems. Recurrent neural networks are different from traditional feed-forward neural networks they have an internal state that can represent context information, they keep information about past inputs.

Structure of a Bidirectional LSTM Network



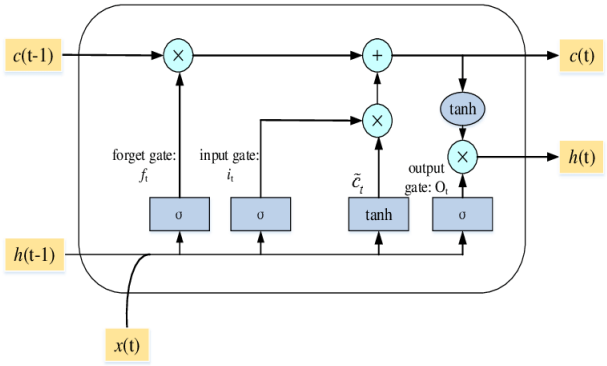
The Long Short Term Memory architecture was motivated by an analysis of error flow in RNNs which found that long time lags were inaccessible to existing architectures, because backpropagated error blows up or decays exponentially.

An LSTM layer consists of a set of recurrently connected blocks, known as memory blocks. Each block contains one or more recurrently connected memory cells and three multiplicative units (the input, output and forget gates) that provide write, read and reset operations for the cells. The net can only interact with the cells via the gates.

The Forget gate is responsible for deciding which information is kept for calculating the cell state and which is not relevant and can be discarded. The ht-1 is the information from the previous hidden state, and xt is the information from the current cell. These are the 2 inputs given to the Forget gate. They are passed through a sigmoid function and the ones tending towards 0 are discarded, and others are passed further to calculate the cell state.

The Input Gate updates the cell state and decides which information is important and which is not. As forget gate helps to discard the information, the input gate helps to find out important information and store certain data in the memory that relevant. ht-1 and xt are the inputs that are both passed through sigmoid and tanh functions respectively. tanh function regulates the network and reduces bias.

Structure of LSTM unit



All the information gained is then used to calculate the new cell state. The cell state is first multiplied with the output of the forget gate. This has a possibility of dropping values in the cell state if it gets multiplied by values near 0. Then a pointwise addition with the output from the input gate updates the cell state to new values that the neural network finds relevant.

The output gate is the last one and decides what the next hidden state should be. ht-1 and xt are passed to a sigmoid function. Then the newly modified cell state is passed through the tanh function and is multiplied with the sigmoid output to decide what information the hidden state should carry.

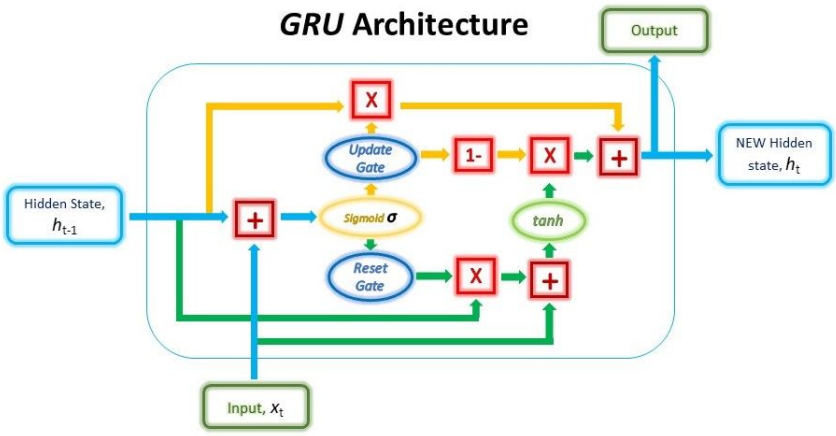
Gated Recurrent Unit

GRUs are improved version of standard recurrent neural network similar to LSTM. To solve the vanishing gradient problem GRU uses update gate and reset gate. These are two vectors which decide what information should be passed to the output. They can be trained to keep information from long ago, without washing it through time or remove information which is irrelevant to the prediction.

An interesting thing about GRU is that, unlike LSTM, it does not have a separate cell state (Ct). It only has a hidden state (Ht). Due to the simpler architecture, GRUs are faster to train. At each timestamp t, it takes an input Xt and the hidden state Ht-1 from the previous timestamp t-1. Later it outputs a new hidden state Ht which again passed to the next timestamp.

Now there are primarily two gates in a GRU as opposed to three gates in an LSTM cell. The first gate is the Reset gate and the other one is the update gate.

Structure of GRU unit



The Reset Gate is responsible for the short-term memory of the network (Ht). Similarly, we have an Update gate for long-term memory.

4. Metrics and Experimental Results

In our pipeline we implemented TF-IDF vectorization with bi-grams and then logistic regression with 10 fold cross-validation. We calculate overall accuracy and probabilities.

For the NN part before feeding our texts in the neural networks we tokenize it in order to transform text into integers and add an embedding layer to the model. We calculate train, validation and test accuracy and loss.

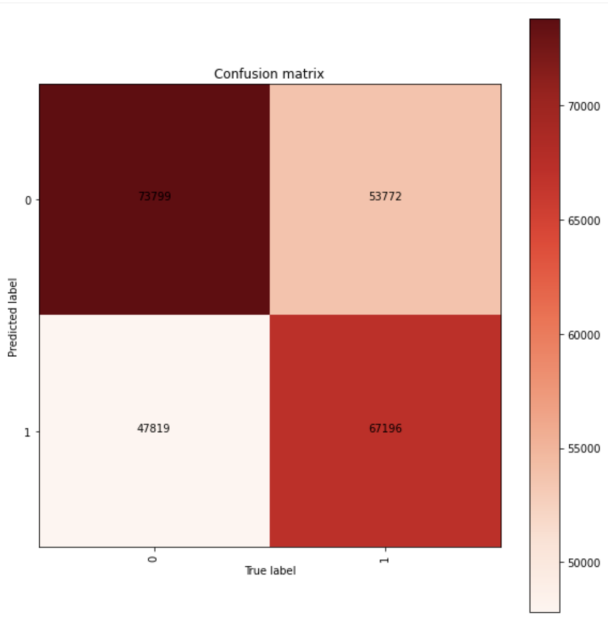
4.1 Logistic Regression

The final overall score was 0.5812, We can print an example of the probabilities that the label is 0 or 1 given for the first 10 rows and the prediction, and see that the prediction is correct in 8 cases out of 10.

```
[[0.5531078  0.4468922 ]
 [0.47618354 0.52381646]
 [0.58630434 0.41369566]
 [0.49201794 0.50798206]
 [0.35039683 0.64960317]
 [0.59389022 0.40610978]
 [0.52124588 0.47875412]
 [0.51103635 0.48896365]
 [0.51438967 0.48561033]
 [0.70416118 0.29583882]]
469600    0
639137    0
240293    0
702254    1
889040    1
118721    0
947749    1
112289    1
564578    0
1020      0
Name: label, dtype: int64
```

Weight[?]	Feature
+2.269	sarcasm
+1.531	people
+1.439	women
+1.293	racist
+1.249	sarcastic
+1.183	woman
+1.146	police
+1.109	rape
+1.041	racism
+1.020	fault
+1.009	clearly
+0.989	government
+0.986	white
+0.943	female
+0.901	children
+0.881	apparently
+0.877	muslim
+0.868	gay
+0.864	shooting
… 2725 more positive …	
… 2256 more negative …	
-0.896	cat

We can show also the weights of every word in our model and see which are the words that account for more sarcasm. We can plot the confusion matrix and see that we have 73799 True Positive, 53777 False Negative, 47829 False Positive and 67196 True Negative



We have tried to implement the logit also considering the subreddit category of the parent comment. The overall score becames 0.5973 if we take in account subreddits as features, so the score has slightly improved.

4.2 Neural Networks

For the training phase we have used binary crossentropy loss and Adam optimizer with 0.001 learning rate. The architectures that we have trained and tested are the followings:

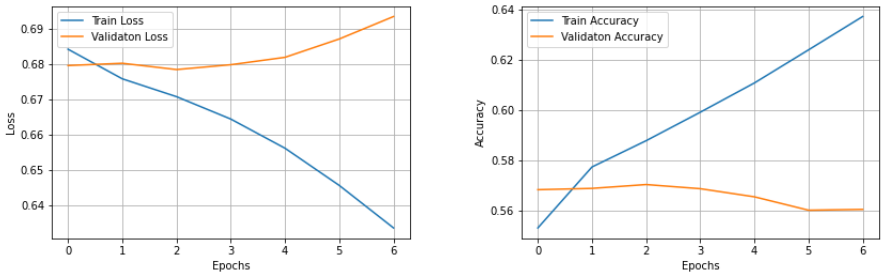
CNN model with 3 convolutional + maxpooling layers with 32,64 and 128 filters with dimensions 5x5 and 3x3, 4 dense layers and 0.2 and 0.5 dropout layers.

LSTM model with 64 cells, 0.4 and 0.5 dropout layers and 2 fully connected layer.

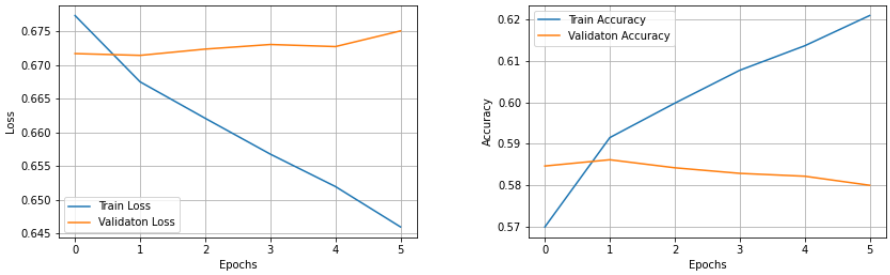
GRU model with 128 cells, 0.5 dropout layer and 3 dense layers.

We have used early stopping with patience 4 for our training and obtained the following results:

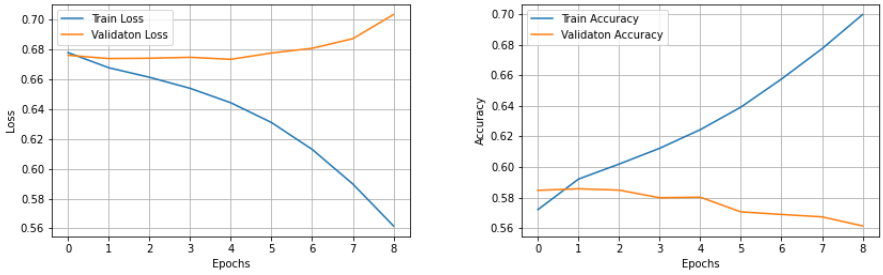
CNN



LSTM



GRU



The training pattern is basically the same for the three models, there is immediately some overfitting and they have also similar validation accuracy but the LSTM one is a bit higher.

We also use a test set to evaluate the best algorithm obtaining 0.5622, 0.5817 and 0.5625 as test accuracy and again LSTM model performed better than the other types of neural network.

5. Conclusions and Remarks

We have tried to implement different models in order to achieve our objective but we haven't reached huge results, in fact our maximum score is 0.5973 with Logistic regression and 0.5817 for the LSTM model.

We are not so satisfied with the results but the task was quite difficult. If we had to choose between our two best models we will choose the logistic regression because is a simpler model than the neural network one.

If we wanted to go deeper in this analysis we could have compared different types of preprocessing (stemming, lemmatization, POS tagging, topic modeling) or word embedding (GloVe and Word2Vec).

We could try other types of classification algorithms like Support Vector Machines and Naive Bayes Classifier for example. Another improvement could be to build a grid search for neural network architecture (n of filters, filters dimensions, n of convolutional layers, n of dense layers different droput) and hyperparameters (optimizer, learning rate); It will be surely time consuming but maybe effective to find the best balance in the architecture.