Boris Alarcon

Sentiment Analysis with TensorFlow

Machine Learning Spring 2024

Table of Contents

Sentiment Analysis

Given an array of textual data extracted from user entries, determine the emotion of such text classifying it as positive, negative, or neutral.

APPLICATIONS

- User content filtering
- Commercial rating systems
- Stock analytics

Common Approaches

Rule Based

Determining scores for every word to then determine the overall score of a text.

Word	Score
Good	0.5
Great	0.8
Terrible	-0.8

Machine Learning Based

- **Word embedding**
- **Logistic regression**
- **Neural network**
- **Random forest**
- **Support vector machine**

Learning sentiment based on use and order of some words, not meaning/value to individual words.

Dataset and Approach

Dataset:

IMDb movie reviews with sentiment labels

imbd.npz from keras.datasets.imdb

+

Word Index

imdb_word_index.json from keras.datasets.imdb

	Reviews	Sentiment
0	‘<START this film was just brilliant casting lo...’	Positive
1	‘<START big hair big boobs bad music and a gian...’	Negative
2	‘<START this has to be one of the worst films o...’	Negative

word	id
'fawn’	34701
'tsukino'	52006
'nunnery'	52007

Steps Overview

1. **Preprocessing**
   Splitting and Encoding
2. **Preparing Neural Network Architecture**
   Word embedding, GAP, ReLU
3. **Training Model**
4. **Testing Model**

Keras & Neural Network

Preprocessing

Reviews	Sentiment
‘<START this film was just brilliant casting lo...’	Positive
‘<START big hair big boobs bad music and a gian...’	Negative
‘<START this has to be one of the worst films o...’	Negative

Reviews

X1	X2	X3	X4	...	Xn
<START	this	film	was	...	
<START	big	hair	big	...	
<START	this	has	to	...	

- Split
- Problem: different *n* size

X1	X2	X3	X4	...	Xn	Sentiment
1	11	19	13	...	0	1
1	191	1150	191	...	0	0
1	11	44	5	...	0	0

- Encode
- Padding (limiting *n*=500)

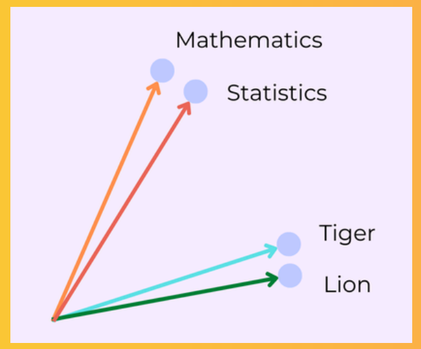
Preparing Neural Network Architecture

Using word embedding, global average pooling layer (GAP), ReLU, and sigmoid layers.

Word Embedding Layer

Learning to represent words as vectors.

This way, the algorithm will learn that words have different structures, meanings, and strengths (among many other things).



Global Average Pooling Layer

Taking the global average of all vector words in a Review.

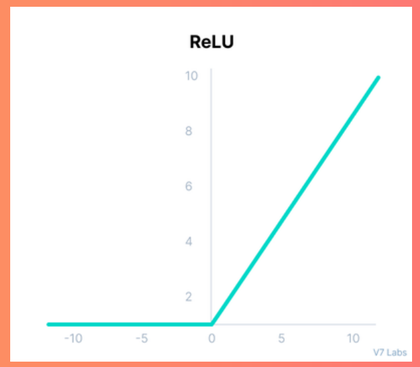
Parameters are reduced when averaging all features. This then reduces the chances of overfitting by simplifying the data.

big:	<1.3, 1.1, 0.4>
hair:	<0.8, 1.6, 0.2>
big:	<1.3, 1.1, 0.4>
boobs:	<1.2, 0.5, -1.3>
bad:	<0.2, 0.7, -2.8>
music:	<1.8, 0.8, 0.8>
GAP:	**<1.1, 0.97, -0.4>**

Dense Layer: ReLU

Rectified Linear Unit

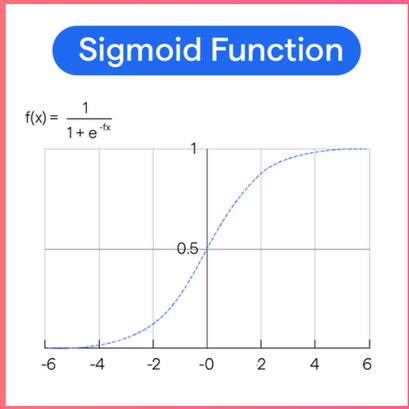
Introduces nonlinearity and deals with the vanishing gradients.



Dense Layer: Sigmoid

Sigmoid activation function

Makes it useful to easily classify the predicted value, while also representing it as a probability.



```
>>> studied = 'True'
>>> if studied:
...     print('Now, you are
...           qualified to
...           review my code.')
```

Thank you!

Boris Alarcon

Computer Science Junior at
Loyola University of New Orleans

Sources:

- https://www.youtube.com/watch?v=hprBCp_UJN0
- https://keras.io/api/datasets/imdb/