MACHINE LEARNING PROJECT

Sentiment Prediction expressed in IMDB reviews

Title

Sentiment Prediction expressed in IMDB reviews

Introduction

Sentiment Analysis or opinion mining is one of the most sought after topics of both Machine Learning and Deep Learning. Sentiment refers to the sequence of words and is usually associated with an opinion or emotion. Analysis refers to the process of looking at the data and making inferences; in this case using Machine Learning to predict whether a movie review is positive or negative. In this study, I develop a classifier that will take as input a user generated movie review and automatically classify that review as positive or negative. To this end I experimented with a wide variety of models including Naïve Bayes and Artificial Neural Network.

Dataset and Features

The dataset used for this task was collected from Large Movie Review Dataset which was used by the AI department of Stanford University for the associated publication. The dataset contains 50,000 training examples collected from IMDB where each review is labeled with the rating of the movie on scale of 1-10. As sentiments are usually bipolar like like/dislike, I categorized these ratings as either 1 (positive) or 0 (negative) based on the ratings. If the rating was above 5, I deduced that the person liked the movie otherwise he did not.

For preprocessing, I first created a copy of the dataset, but in lowercase. I then created several variations of the lowercase dataset using different combinations of preprocessing methods. The four preprocessing methods I used were removing punctuation, tokenization, stemming, and removing stop words.

I split this dataset 60/20/20, with 60% becoming the training set, 20% becoming the validation set, and the last 20% becoming the test set.

Dataset link : https://www.kaggle.com/lakshmi25npathi/imdb-dataset-of-50k-movie-reviews

Methods used for model creation

1. Naive Bayes

Naive Bayes is a generative learning model that takes the distribution of words given labels from the training set to compute the probability of a test set statement having a positive or negative label. The distribution from the training set is usually modified somewhat using Laplace Smoothing, which deals with unseen vocabulary in the training set. The exact equations for calculating the distributions are as follows:

$$\phi_{j|y=1} = \frac{1+\sum_{i=1}^{n} 1\{x_j^{(i)}=1 \wedge y^{(i)}=1\}}{2+\sum_{i=1}^{n} 1\{y^{(i)}=1\}}$$

$$\phi_{j|y=0} = \frac{1+\sum_{i=1}^{n} 1\{x_j^{(i)}=1 \wedge y^{(i)}=0\}}{2+\sum_{i=1}^{n} 1\{y^{(i)}=0\}}$$

$$\phi_y = \frac{\sum_{i=1}^{n} 1\{y^{(i)}=1\}}{n}$$

I also read that a Naive Bayes could theoretically run with fractional counts, so I also ran a variant using tf-idf scores instead of word counts.

$$\theta_j := \theta_j + \alpha \left(y^{(i)} - h_\theta(x^{(i)})\right) x_j^{(i)}$$

1. Artificial Neural Networks

Artificial Neural Networks is an algorithm that learns a set of weights and bias for each layer l to get the hypothesis as:

$$z^{[\ell]} = W^{[\ell]} a^{[\ell-1]} + b^{[\ell]}$$

$$a^{[\ell]} = g^{[\ell]}(z^{[\ell]})$$

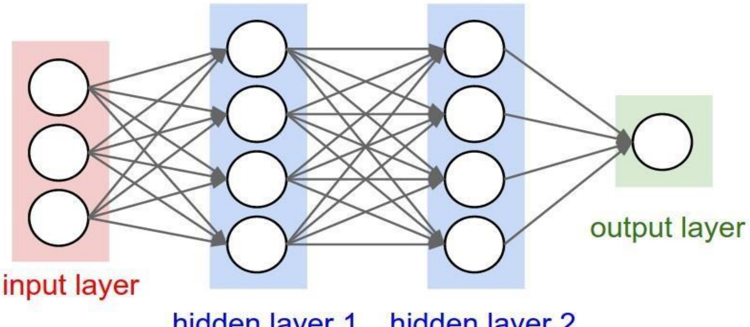
For the training part back propagation algorithm is used as:

$$\delta^{[\ell]} = (W^{[\ell+1]\top} \delta^{[\ell+1]}]) \circ g'(z^{[\ell]})$$

$$\nabla_{W^{[\ell]}} J(W,b) = \delta^{[\ell]} a^{[\ell-1]\top}$$

$$\nabla_{b^{[\ell]}} J(W,b) = \delta^{[\ell]}$$

In this model, I have trained a neural network having two hidden layers with neurons in input layers= length of dictionary, neurons in first hidden layers=16, neurons in second hidden layer=16 and neurons in output layer=1. I used ReLU as activation function in both the hidden layers and Sigmoid function as the activation function of the final output layer.



Implementation and Results

	Accuracy	F1	Precision	Recall
Naïve Bayes(tf-dif)	0.874	0.872	0.891	0.854
Naïve Bayes(Word Count)	0.866	0.864	0.880	0.848
Neural Network(ANN)	0.907	0.908	0.906	0.909