TUHH
Hamburg University of Technology

Research Project

David Engelhard

Understanding and Improving Neural Network Classification

November 25, 2018

supervised by:
Prof. Dr. Sibylle Schupp
Anna Lainé

Hamburg University of Technology (TUHH)
Technische Universität Hamburg-Harburg
Institute for Software Systems
21073 Hamburg

STS
Software
Technology
Systems

Sworn declaration

I declare under oath that I have prepared the paper at hand independently and without the help of others and that I have not used any other sources and recourses than the ones stated. Parts that have been taken literally or correspondingly from published or unpublished texts or other sources have been labeled as such. This paper has not been presented to any examination board in the same or similar form before.

_____

Date, signature

Abstract

Neural networks are powerful tools for dealing with classification problems. However the interior of a neural network is hardly understandable for humans. Usually, the only indication of the quality is the accuracy of a neural network. So improving a network consists of trying out different approaches and compare their accuracy. This thesis gives an approach on how to tackle the problem of improving by first understanding what the neural network does and comparing this understanding with slightly different neural networks. This way the improvement can not only be measured by the accuracy but also by more concrete decisions made by the neural network.

Contents

List of Figures

1 Introduction

In computer science there are many topics where neural networks are applicable. Neural networks are particularly well suited for the problems where input needs to be classified into predefined classes. The most commonly used applications are natural language processing and image recognition. There are many big data sets of those fields that are available in the internet. Even though it is used frequently, there are still problems when using neural networks. In this thesis I give an approach on how to tackle the problems of understanding what a neural network has learned and how it could be improved using that knowledge. First to tackle the problem of understanding neural network classification, the paper "What does my classifier learn? - A visual approach to understanding natural language text classifiers"[1] is used and extended. It gives an approach on how to get some understanding on what a neural network learned. The paper works on neural networks for text classification where the input is easily comprehensive for humans. This thesis extends the paper "What does my classifier learn?"[1] by giving conclusions to the results and giving an approach on how to improve the neural network. This can only be done in fields where humans could classify the examples themselves and know what should be an important classification criteria for the neural network. To tackle the problem of improving the neural network classification, different neural networks were trained and the accuracy measured. With both parts combined one can find regularities in behavior of neural networks by applying different changes regarding their construction and training.
In the next chapter a basic setup for training neural networks for text classification is presented. This is used in the following to explain how to understand neural network classification. Also it will serve as a starting point for the different changes, that were applied in the experiments. Using the same starting point simplifies the comparison of the experiment results. Combining the understanding of neural network classification with the results of the experiments will lead to the chapter of how to improve neural network classification.

2 Basic setup for neural network text classification

This chapter will introduce the decisions and the data that the project started with. Due to the fact that in the paper "What does my classifier learn?"[1] a convolutional neural network for text classification was used, the setup was reused as the starting point.

2.1 Data set

The datas et used throughout this whole thesis is a extract of the IMDB movie review data base. The data was collected and used in the paper "Learning Word Vectors for Sentiment Analysis"[2] in 2011. The paper discusses different approaches for sentiment analysis in natural language processing. The data set consists of 50,000 reviews and is split in 50% training and 50 % evaluation data. In this thesis the split is preserved to make a comparison to the other papers that used the data set possible. All reviews in the data set have a score between 1 and 10. The reviews are divided into two classes. The classes are positive and negative. All reviews with a score less than 5 belong to the negative class and all reviews with a score higher than 6 belong to the positive class. There are no reviews with scores 5 or 6 in the data set. The length of the reviews range from 6 to over 2000 words.
The best approach in the paper "Learning Word Vectors for Sentiment Analysis"[2] achieved an accuracy of 88.89% on the data set. The data set was also used in a kaggle competition. The winners achieved an accuracy of 99%.

2.2 Word embedding

In the domain of text classification the representation of the text is an important factor to the neural networks performance. The texts need to be transformed into a vector representation in order to process it further by a neural network. This field is called word embedding and the following approaches were considered for the basic setup.
The straight forward approach to this problem is to take a vector with the size of the number of words that exist. So each entry represents one particular word. This is a one hot representation where the vectors only contain a one in one entry and zeroes in all others. There are a few downsides to this representation:

- the embedded word vectors are very large
- it is hard to extend it with new unknown words
- there is no relation between any words
- the sentiment is not captured

Another approach to word embedding is to represent the words by vectors of a fixed length with entries of real values called feature vector. To tackle the downsides of the one hot vector representation the vectors representing a word must be learned. There are models called Word2vec that do such learning using neural networks. Word2vec was developed by a research team from Google[4]. The result of a trained Word2vec model is a dictionary where each word is mapped to a feature vector of fixed length. The goal of the training is that similar words are represented by similar vectors. For example, very similar words have a small distance between their feature vector representations. This helps a lot when doing sentiment analysis using neural networks. A neural network can still give good results even when a word never occurred in the training data. It is enough that similar words were trained.
Most of the training models of Word2vec are unsupervised and follow the rule "Similar words occur in similar contexts"[3]. In order to reliably find such similarities a big data set is needed. There is a pre trained model that was released by Google. It contains 3 million words where each word is embedded by a 300 entry feature vector.
To use Word2vec, it can be trained during the neural network training or a pre-trained model can be used. I tried out both ways and compared the results. When training Word2vec while neural network training, the context of the data set can be covered which is not possible for a pre-trained model. The Word2vec training is then part of the neural network that basically extends it by another layer. The problem I encountered is the size of the data set. Some of the words that are in the evaluation data are not covered in the training data. So when evaluating the neural network uses an untrained embedding for those words. When using the pre-trained Word2vec model of Google this problem does not occur. Due to Googles very large data set (basically the whole internet) and their computational power, all 3 million words in that model are trained. Using the pre-trained model lead to an increase in training time, but also in an increase of accuracy when evaluating. The training time mainly increases due to two factors. First, the neural network is smaller by taking out the embedding training so it needs more training cycles to converge to a fix point. Second, Googles Word2vec model is very big. To load and use such a big dictionary cost some performance. None the less, I chose to use the pre-embedded model of Google because of its large number of words and its quality.

2.3 Neural network structure

The structure of the neural network is close to the structure given in the paper "What does my classifier learn?"[1]. It is a common structure for text classification that is also used in other literature. The structure will be explained layer by layer.

2.3.1 Word embedding

As written in the previous section, the word embedding could be trained as part of the neural network. In my case I chose the pre-trained Google model with feature vectors of size 300. That is why the embedding is not a trainable part of the neural network. To

be able to process the embedded matrix in the following layer, it needs to have a fixed number of rows which represents the number of words. The review length ranges from 14 to 2697, but most of the reviews have less than 300 words, so I chose the fixed length to be 300. All reviews that are shorter will be extended with zero values and all reviews that are longer will be cut at 300 words.

$embedding : String^{w} \rightarrow \mathbb{R}^{l \times emb}$ where
$w = \#\text{words}$, $l = 300$(max review length) and $emb = 300$(embedding size)

2.3.2 Convolutional layer

Convolutional layers use sliding windows over the input. They apply very well to two dimensional input such like the embedded sentence. The sliding window provides the possibility of learning the correlation of words following each other. This applies for example to negation. The neural network can learn that the review "This movie was not good, it was actually quite bad" has a negative sentiment, because the word "good" followed the word "not". There are other approaches that do not use convolutional layers. For example, the continuous bag of words (CBOW) squishes all words of a sentence into one vector[3]. All words are influencing the result, but the order is neglected. This leads to the problem that the sentences "The movie was not bad, it was actually quite good" and "The movie was not good, it was actually quite bad" will result in the output using CBOW.
A convolutional layer consists of multiple filter sets, that define the sliding window. Each filter set has a size, in this case a fixed number of consecutive words it filters. Each filter in the filter set has its own trainable weights and biases. Basically, the more filters there are, the more characteristics the convolutional layer can learn. I chose to have 3 filter sets with filter lengths of 3, 4 and 5 consisting of 64 filters each. This means the layers will look at up to 5 consecutive words. A relation between words cannot be learned if they are are farther away from each other.

For each filter set:
$conv : \mathbb{R}^{l \times emb} \rightarrow \mathbb{R}^{l - flen + 1 \times fnum}$ where
$flen \in \{3, 4, 5\}$ (filter length) and $fnum = 64$ (number of filters)

2.3.3 Max-pool-layer

The max-Pool layer is applied to each filter set and takes the maximum value for each filter. This results in a vector that is just as long as the number of filters and it takes only the most influencing part of each filter into account. It is commonly used in combination with the convolutional layer to effectively shrink the big number of outputs that is generated by the convolutional layers. That way every filter can focus on specific occurrences of feature vector combinations and will put out a high number is that combination was found.

For each filter set:
$maxpool : \mathbb{R}^{l-flen+1 \times fnum} \rightarrow \mathbb{R}^{fnum}$

2.3.4 Concatenation layer

The concatenation layer just puts the 3 vectors together so they can be processed by the last layer.

$concat : \mathbb{R}^{fnum \times 3} \rightarrow \mathbb{R}^{fnum \cdot 3}$

2.3.5 Fully connected output layer

The fully connected output layer is the last layer and has just two outputs namely positive and negative. It is a typical fully connected layer with trainable weights and biases. The highest value of the output defines the calculated classification.

$output : \mathbb{R}^{fnum \cdot 3} \rightarrow \mathbb{R}^{2}$

2.4 Training parameters

The training of the neural network cannot be done by just passing the training data through it once. The training needs to go multiple times through the data to find an optimum using gradient descent. That is why a fixed number of epochs is needed. An epoch is defined as the number of times the training data is run through during training. Another problem is that the whole training data set is too big to run through the training steps all at once, so the training data needs to be split into smaller batches. After several training runs, I set the number of epochs to 10 and the batch size to 128. This leads to a total number of 2000 steps needed to run the training. It took about 1 hour of calculation on my device to finish the training, which is a reasonable time, considering multiple experiments that all need to run through the training.

2.5 Implementation

The implementation of the neural network was done in Python using the TensorFlow framework from Google. "TensorFlow™ is an open source software library for high performance numerical computation. Originally developed by researchers and engineers from the Google Brain team within Google's AI organization, it comes with strong support for machine learning and deep learning and the flexible numerical computation core is used across many other scientific domains."[1]. It is easy to use and updated regularly. Due to its popularity, questions and problems are solved by the community pretty fast.
For a basic structure of my files I chose to use a sample code used in a kaggle class[2]. It

[1] https://www.tensorflow.org/
[2] https://www.kaggle.com/c/cls-text-classification

is divided into multiple files:

- data_helpers: Loads the data set and transforms it into a suitable structure
- text_cnn: The definition of the neural network
- train_cnn: The implementation of the training and saving of the results
- eval_cnn: The implementation of the evaluation of saved neural networks

By the use of many parameters in the files for the neural network and the training, it is easy to tweak the parts that need to change for my experiments.

3 Understanding classification

This section will deal with the problem of understanding neural network classification. Neural networks learn by themselves and the only parts that human can handle easily, are the inputs and the outputs. The node activations, weights and biases of a neural network are just trained numbers that can only be interpreted computationally. The goal is to somehow process those numbers in combination with the input and output to calculate something, that is really understandable. To do so, the paper "What does my classifier learn? - A visual approach to understanding natural language text classifiers"[1] is used, that will be explained in the following section.

3.1 "What does my Classifier learn?"

The authors of the paper "What does my Classifier learn?"[1] approach the problem of understanding neural network text classification to make it more acceptable to the users. Users of software are generally skeptical, especially when the reason behind the decisions are very hard to comprehend. This is the case for neural networks, where all inner states are just numbers, that are hard to link to sentiment or other human understandable representations. Their approach is to give such a representation by calculating the influence of the words on the resulting classification. The words are then colored depending on their influence as seen in the figure 3.1.

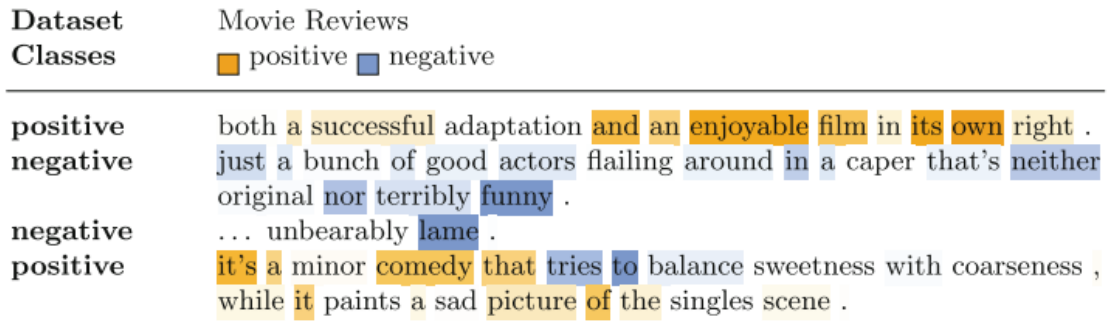

Figure 3.1: Example of coloring depending on the word influences on the classification

In their approach they present a neural network structure that is very close to the one already presented in 2.3. They present a trace back algorithm that returns the influences of words on the classification in the form of an document influence matrix (DIM). The DIM has a column for each word and a row for each class. In the example of the movie reviews there are just two classes - positive and negative. For example, the first row of the DIM contains the influence of the first word of the input on the respective classification. The algorithm calculates an intermediate influence matrices (IIM) for each layer in order to get to the final DIM. The IIM contains the influence of the nodes on the classification. The first IIM that can be calculated is of the output layer. It is just

an diagonal matrix where the values are the activation of the output nodes. Taking the first IIM, the weights and activations of the previous layer one can calculate the next IIM. It follows the principle that a combination of high weights and activation lead to a high influence. In the paper are formulas that can calculate an IIM for each type of layer used. After running through all layers one get the IIM of the embedding layer. It contains influence values for each feature of each word. After summing up the influence values of all features belonging to each word and normalizing it, one gets the DIM.
This approach was applied to their own requirement classification tool and they received very positive feedback for it. The users trusted the system more and can react to the output. Problems in their phrasing or in the neural network could easily be revealed. In my approach I will focus on the problems of the neural network.
To do so I tweaked the algorithm slightly for my purpose. Thus I wanted to use the raw scoring output of the network, I did not apply the softmax function on the output of the last layer. For the traceback algorithm, I chose the first IIM to be the identity matrix. This way the actual classification scores do not influence the calculation of the DIM. The results using this approach were much more fitting for my examples. Because the problem of the movie reviews I deal with in this thesis is a binary problem, I chose to use a vector of the influences instead of the DIM. For every word, subtract the negative influence from the positive influence. This increases the expressiveness of the colored output. For example, coloring a word that is equally high influencing the positive and the negative class. In the end it the word is neutral because it increases both equally, which does not change the distance of the output scores. With the approach from the paper it will either be colored totally positive or negative depending on which one is slightly higher. Using my approach, the word would not be colored because it influence value is close to zero.

3.2 Categorize trained neural networks

From the previous presented coloring approach not only users but also the developers can take advantage of it. In the field of text sentiment classification the developer usually knows what results to expect. For example, negative words lead to negative classification. Or negation using the word "not" usually negates the positivity or negativity of the following word. The developer now has a way of telling if the neural network learned patterns like negation or sentiments like positive and negative. In the following neural networks will be classified into predefined classes using that knowledge.
I chose to look at the problem of over fitting and under fitting. A neural network is over fitting if it learned the training data exactly. For example, when a neural network is very big and there is only a small amount of training data, it can memorize the training data exactly even the noise in the training data. This leads to a high accuracy on the training data, but can lead to very poor results for other data. The neural network does not know how to classify, when the concrete sequences from the training data do not appear in the evaluation data. The creator of a text classification neural network generally wants it to learn patterns. That is why over fitting should be prevented. Under fitting on the other

hand happens, when the neural network is too small to learn all the regularities from the training data. This results in a neural network that can neither model the training data nor generalize to new data.
To classify the neural networks into the classes over fitting and under fitting, I chose to just look at the top influencing words and their surroundings for the classification. For neural networks that are over fitting and under fitting the most influencing words should have no real sentiment in the sense of movie reviews. For the neural networks that are fine between over fitting and under fitting the sentiment of the top words influencing should be clear.

4 Experiments

In the following experiments the classification of neural networks is applied to trained neural networks. The first experiment is a neural network with the structure shown in 2.3. The other experiments are neural networks, that have just one parameter changed. I chose changes from different parts that have an impact on the training, to get an idea of how they influence the quality of the output and where one can get the most optimization. To evaluate the trained neural network, I chose 4 reviews out of the 25000 reviews from the evaluation data. Those 4 reviews were classified incorrectly by some of the trained neural networks. They are found in the appendix starting at page 41. Each experiment will start with a listing containing the properties of the neural network, where the differences compared to the unchanged neural network are colored in red and an explanation why certain changes were chosen. After that the accuracy on the whole 25000 evaluation reviews, followed by each experiment is analyzed using the procedure explained in chapter 3.

4.1 Unchanged neural network

- Filter set sizes: 3, 4 and 5
- Number of filters per filter set: 128
- Use pre-embedded data: True
- Embedding size: 300
- Max number of words: 300
- Number of training epochs: 10
- Training batch size: 128

This network had an accuracy of 88.62% on the evaluation data. The creators[2] of the data set came to an accuracy of 88.89%, so their result is only slightly better. The reasons for this structure are already explained in 2.3. For this experiment, the colored result as described in the paper “What does my classifier learn?” [1] (with the addition of my tweaks described in the end of 3) is shown for each of the 4 evaluation reviews. Words with a colored background are influencing the classification, where the red background stand for negative words and the green background for positive words. The stronger the color, the stronger the influence on the classification. Because of the max number of words restriction, only the first 300 words will be analyzed. The words that were not analyzed with the trace back algorithm, did not influence the classification of the review. Also the final scores computed by the neural network are given for each evaluation review. The higher score decides the classification result. When both values are very close to each other the network could not really decide which class to chose.

For the other experiments the whole reviews are not shown but only the top influencing words that will be compared with the unchanged model. The top influencing words will be attributed with their normalized influence on the classification, where negative values indicate influence in the direction of the negative class and positive vice versa.

Positive review 1 (score 9 of 10)

Classified as positive by the neural network with the scores:

- Positive: 5.169222
- Negative: -1.8755013

" a talented high school graduating senior with a bad attitude is forced to play in the state all star high school football game when he meets and falls for an attractive local girl she helps him realize he has a shot at a 'full ride' scholarship if he plays well br br all too often , these dramas fall into formulaic traps and tell the same old story of a troubled and confused teen full ride 's matt sabo certainly fits this profile , but below the surface is a much more unique individual than we usually see in this genre matt is the center of the action and he is a realistic teenager , both over confident and vulnerable , optimistic and cynical by turns influenced by amy , matt grows into a man of character and heart he , in turn , forms friendships with his teammates , which influences his growth as an athlete and as a team player br br full ride has all the elements we love to see in a movie great acting , admirable characters , exciting sports scenes , poignant drama , and a love story still , while one may have seen these elements in other films , full ride is assisted by performances that are sincere and occasionally , even moving perhaps what 's most impressive about full ride is its sense of reality although the author of the previous comment would seem to disagree , (clearly a disgruntled student who , for quite obvious reasons , received a poor grade in his film class) director mark hoeger grounds the film in a believable situation and location and does a great job of getting down to the grit of what life is like in a small town these "
Rest of the review: "characters are real people rooted in realistic situations , which often create the most compelling entertainment on one level it is a love story , on another it is a character study , and yet another it is a simple football film all of these ideas come together to form a cohesive vehicle"

Top positive words:

1. poignant: 0.327718433910604
2. great: 0.27151082391023706
3. impressive: 0.24050007873735632

Top negative words:

1. formulaic: -0.5163059361365703
2. poor: -0.4320636762217847
3. bad: -0.2589653122190472

There are many positive words in the review which in the end influenced the positive classification. Even though it is classified as positive, there are the top three strong negative influencing words, that are stronger influencing than the top positive words. The words "formulaic" in this review is indeed meaning a bad kind of movie profile, but in the same sentence the author writes, that it is different and unique compared to those movies. This figure of speech is not covered by the neural network. The other negative influencing words are not based on his opinion of the movie but a character in the movie and another comment author. For a better classification quality, those negative words should not influence the classification, when they are referring to something else than the sentiment regarding the movie.

Positive review 2 (score 9 of 10)

Classified as negative by the neural network with the scores:

- Positive: 1.495384
- Negative: 2.389628

" if you keep rigid historical perspective out of it , this film is actually quite entertaining it 's got action , adventure and romance , and one of the premiere casting match ups of the era with errol flynn and olivia de havilland in the lead roles as evident on this board , the picture does n't pass muster with purists who look for one hundred percent accuracy in their story telling to get beyond that , one need only put aside the history book , and enjoy the story as if it were a work of fiction i know , i know , that 's hard to do when you consider custer 's last stand at the little big horn and it 's prominence in the history of post civil war america so i guess there 's an unresolved quandary with the picture , no matter how you look at it br br there 's a lot to take in here though for the picture 's two hour plus run time custer 's arrival at west point is probably the first head scratcher , riding up as he does in full military regalia the practical joke by sharp (arthur kennedy) putting him up in the major 's headquarters probably should have gotten them both in trouble br br ironically , a lot of scenes in this military

film play for comedy , as in custer 's first meeting with libby bacon , and subsequent encounters that include tea reader callie (hattie mcdaniel) i had n't noticed it before in other films , but mcdaniel reminded me an awful lot of another favorite character actor of mine from the forties , mantan moreland so much so that in one scene it looked like it might have "
Rest of the review: "been moreland hamming it up in a dress with that in mind , the owl scene was a hoot too br br as for flynn , it 's interesting to note that a year earlier , he portrayed j e b stuart opposite ronald reagan 's depiction of general custer in santa fe trail , both vying for the attention of none other than olivia de havilland in that film , reagan put none of the arrogance and flamboyance into the character of custer that history remembers , while in flynn 's portrayal here it 's more than evident but it does n't come close to that of richard mulligan 's take on the military hero in 1970 's little big man let 's just say that one was a bit over the top br br the better take away the picture had for me was the manner in which custer persevered to maintain his good name and not gamble it away on a risky business venture that and his loyalty to the men he led in battle along with the discipline he developed over the course of the story most poignant was that final confrontation with arch rival sharp just before riding into the little big horn , in which he declared that hell or glory was entirely dependent on one 's point of view earlier , a similar remark might have given us the best insight of all into custer 's character , when he stated you take glory with you when it 's your time to go"

Top positive words:	Top negative words:
1. favorite: 0.3121136405350581	1. awful: -0.8066658328385469
2. entertaining: 0.25262297840563913	2. an: -0.1309286348696559
3. enjoy: 0.1599456965109478	3. joke: -0.11999123236982584

Even though "awful" is a sentimental word, it seems like it is rated out of context. For the reader of the review it is clear that the word awful is not meant as a negative word. The author just meant that one actor really reminded him of another character he liked. In this case "awful" is stronger influencing the negative classification than all three most positive influencing words combined and this leads to a wrong classification.

Negative review 1 (score 3 of 10)

Classified as positive by the neural network with the scores:

- Positive: 3.0199063
- Negative: 1.6939285

" while it 's true that the movie is somewhat interesting , the execution leaves a lot to be desired (much like blood orgy of the leather girls

, i spit on your grave , and born in flames , all superior) i do n't think it 's not porn , but porn is in the eye of the beholder if it functions as porn for somebody , who am i to say that he she is wrong ? i was rather puzzled by the statement in winkimation 's generally thoughtful review (such a shame) that for once we actually see men 's faces when they come a few years ago i did occasional freelance reviews for an adult mag and i recall seeing plenty of men 's faces when they came i think this is probably more common when the film features on of the few male porn stars (and especially when that male is the director) though i unsurprisingly ca n't refer to any specific titles , i know that there are some instances in ron jeremy 's , uh , work i also do n't know that i 'd agree that a man is necessarily showing vulnerability in his face when coming "

Top positive words:

1. thoughtful: 0.4432305422049005
2. superior: 0.2679338642532324
3. interesting: 0.2060398459257795

Top negative words:

1. director: -0.3201376400140172
2. wrong: -0.2897758875613409
3. n't: -0.2579769233248799

This is a hard review to classify. The most important aspect is basically the first sentence. It says the movie is interesting but leaves a lot to be desired. The rest of the review discusses one feature of the movie and whether or not it is good thing. Analyzing the positive words, one sees a recurring problem of the neural network. All positive words except "interesting" are referred to something else than the movie.

Negative review 2 (score 2 of 10)

Classified as positive by the neural network with the scores:

- Positive: 2.9563015
- Negative: 0.2557457

" water shows the plight of indian widows in the late 1930s , says in the end that the problem still exists largely by giving statistics in the end , refers to gandhi several times in the movie before finally having a scene depicting him and does nothing extra ordinarily innovative or new in the movie yes , the cinematography is pretty impressive but that cannot be the soul of any movie for me br br india has had several problems like

many other nations but it has got rid of many of these problems at large what if a movie is made on racism in america in a particular year which ends with 'x number of americans still experience racism today' br br a) how would it be relevant , and , b) how would it be some thing so extra ordinary being depicted in cinema br br a view i read from a deepa mehta interview was that this movie is being interpreted as a voice for the marginalised every where from reviews i read every where , the common thing i am hearing is how the director did a great job and was brave in bringing a problem to the world the movie is more about a specific problem a society faced (and has got rid of through reforms at large) br br i do not see any thing earth shattering about the movie moreover , the movie lacked soul and shifted between the plots of chuiyya and kalyani sarala , the young sri lankan actress , portrayed the role of chuiyya superbly and that was the only thing which impressed me about the movie , sadly "

Top positive words:

1. superbly: 0.5922590493270259
2. great: 0.24924683092695382
3. impressive: 0.21345357999241843

Top negative words:

1. cinematography: -0.3078312919359673
2. lacked: -0.2968981120248911
3. nothing: -0.2925764732627304

As a reader of this review, one can clearly see that the negative aspects of this review are superior to the few positive aspects. However, the positive aspects have a much higher impact for the neural network classification. On the other hand, the word "cinematography" is rated really negative despite its positive context. In the end it was classified as a positive review, even though the author rated it 2 of 10.

Summary

The classification has problems to deal with the contexts of the words. Generally the highlighted words are sentimental, but some of them are not. For example the words "cinematography" and "director" influenced the classification negatively despite the fact, that they were used in a neutral or positive context. The neural network seems to be slightly over fitting for these words. The following experiments will be compared with this result and refer to it as the "unchanged model" and the "unchanged neural network".

4.2 Add additional training data

This is the first and easiest change I could think of. Adding the wrong classified reviews to the training data will definitely lead to a neural network that can classify those correctly.

- Filter set sizes: 3, 4 and 5
- Number of filters per filter set: 128
- Use pre-embedded data: True
- Embedding size: 300
- Max number of words: 300
- Number of training epochs: 10
- Training batch size: 128
- Added the 4 evaluation reviews to the training data

The trained network has an accuracy of 88.69% on the evaluation data which is negligibly close to the accuracy of 88.62% of the unchanged model.

Positive review 1 (score 9 of 10)

Classified as positive by the neural network with the scores:

- Positive: 4.6463237
- Negative: -2.860028

Top positive words:

1. poignant: 0.4089174118155082
2. great: 0.3163259805522882
3. job: 0.252985402352254

Top negative words:

1. formulaic: -0.4831930546672574
2. poor: -0.42217156247802606
3. bad: -0.2283037065699306

The result for this review did not change much. It was classified correctly before and even the top 3 words for the both classes only changed slightly. The word "job" is now more positive than the word "impressive" which was the third most positive word for the unchanged model.

Positive review 2 (score 9 of 10)

Classified as positve by the neural network with the scores:

- Positive: 2.9493341
- Negative: -1.4915553

Top positive words:

1. entertaining: 0.4962669643670855
2. favorite: 0.30036749264724655
3. quite: 0.20237636023554997

Top negative words:

1. awful: -0.6052359629273601
2. joke: -0.15146485616791763
3. actor: -0.13572064046001067

This review classification has a totally different result than for the unchanged model. As expected, it is classified correctly. The word "entertaining" is now much stronger positivly than in the unchanged model and the word "quite" that is before "entertaining" is quite positive as well. The previously totally negative word "awful" is a lesser negative so the positive words can surpass it, which leads to the positive classification.

Negative review 1 (score 3 of 10)

Classified as negative by the neural network with the scores:

- Positive: -2.3516521
- Negative: 1.8305578

Top positive words:

1. thoughtful: 0.4659167436418613
2. years: 0.16415934108320732
3. i: 0.13692122481675312

Top negative words:

1. director: -0.3102878709228293
2. spit: -0.28186183816122834
3. orgy: -0.22471608650556887

Compared to the unchanged neural network, this top words are lesser sentimental. Because this review is very hard to classify in the first place, the neural network learned some words to be negative that do not regular appear in reviews, so that the classification in the end will be negative.

Negative review 2 (score 2 of 10)

Classified as negative by the neural network with the scores:

- Positive: -1.8948686
- Negative: 1.855837

Top positive words:

1. superbly: 0.4539857536856387
2. impressive: 0.38603067055097745
3. great: 0.2923597363774577

Top negative words:

1. lacked: -0.34871510023946756
2. cinematography: -0.26280298770216004
3. director: -0.202536121120539

The top words and their influences are very close to the top words and influences of the unchanged model. Still this neural network classified the review as a negative review. There must be more words that influence the negative classifications a bit more than in the unchanged model, but still not strong enough to appear in the top 3 words.

Summary

The evaluation reviews were all classified correctly, but for the ones that were not classified correctly by the unchanged model, the top influencing words seem to be less sentimental. The neural network tried to find rules for the regularities in the evaluation reviews that were not present in other reviews.

4.3 Train word embedding

The ways of how to approach word embedding are discussed in 2.2. This experiment will help to determine the influences of trained embedding compared to a big pre embedded vocabulary used in the unchanged model.

- Filter set sizes: 3, 4 and 5
- Number of filters per filter set: 128
- Use pre-embedded data: False
- Embedding size: 128
- Max number of words: 300
- Number of training epochs: 10
- Training batch size: 128

The trained network has an accuracy of 87.38% on the evaluation data which is slightly lower than the accuracy of 88.62% of the unchanged model.

Positive review 1 (score 9 of 10)

Classified as positive by the neural network with the scores:

- Positive: -0.11361347
- Negative: -5.516946

Top positive words:

1. great: 0.315057747574907
2. job: 0.31468455105502285
3. great: 0.1985835398709619

Top negative words:

1. poor: -0.34283121286916146
2. bad: -0.3411021566715231
3. scenes: -0.10170057102682106

The words "great", "poor" and "bad" are sentimental words and fit quite well in the sentiment classification. But it deals with the same problem as the unchanged model, for example dealing with the context of the word "poor" in this review.

Positive review 2 (score 9 of 10)

Classified as negative by the neural network with the scores:

- Positive: -2.4545689
- Negative: -2.262425

Top positive words:

1. best: 0.15942598012127357
2. favorite: 0.12746066112924112
3. portrayal: 0.06906213715938016

Top negative words:

1. awful: -0.39219290803630574
2. if: -0.08009185997725579
3. name: -0.07457507147273561

The scores of the classes for this classification are very close to each other. The only word that is also in the top influencing words of the unchanged model is "awful", but with a lower influence. All other words have an even lower influence on the classification. This shows that the neural network was very indecisive for most of the words.

Negative review 1 (score 3 of 10)

Classified as positive by the neural network with the scores:

- Positive: -2.617371
- Negative: -4.8306937

Top positive words:

1. and: 0.10196201087701248
2. plenty: 0.09350735493440133
3. i: 0.0847242440127226

Top negative words:

1. any: -0.11598067026754058
2. much: -0.08528370184889994
3. they: -0.07672489996278403

In this result there is not even one sentimental word for this review. Even though it performs worse than the unchanged model regarding the sentimental value of the words, it is still not worse regarding the calculated classifications. Both are wrong with a distance between the positive and negative scores from about 2.

Negative review 2 (score 2 of 10)

Classified as negative by the neural network with the scores:

- Positive: -3.10579
- Negative: -3.5131009

Top positive words:

1. great: 0.5848658711277873
2. job: 0.365782412389305
3. a: 0.1654192597174395

Top negative words:

1. nothing: -0.3154974035451103
2. only: -0.1496556165354698
3. not: -0.11701074729626634

The neural network was very indecisive for this review. The words "nothing" and "great" are also very influencing for the classification in the unchanged model. Missing are words like "superbly" and "lacked", that have high sentimental value but do not influence this sentiment classification very much.

Summary

Regarding the accuracy this approach seems to be only slightly worse than the unchanged model. Looking on the influencing words for the classifications in detail show that not all words that are sentimental are learned in the training. This leads to some very close scores for the classification and shows that the neural network might be under fitted. That would make sense because there is not enough training data to fill the whole vocabulary with all the words that appear in the evaluation data.

4.4 Change neural network structure

The following changes on the neural network structure are just changing the number of filters per filter set in the convolutional layer. This change is highly influencing the number of nodes of the neural network and could easily be changed by just changing one parameter in the training implementation. This experiment should find a correlation between node number and over fitting or under fitting.

4.4.1 Increase number of filters

- Filter set sizes: 3, 4 and 5
- Number of filters per filter set: 256
- Use pre-embedded data: True
- Embedding size: 300

- Max number of words: 300
- Number of training epochs: 10
- Training batch size: 128

The trained network has an accuracy of 88.62% on the evaluation data which exactly the same as the accuracy of 88.62% of the unchanged model.

Positive review 1 (score 9 of 10)

Classified as positive by the neural network with the scores:

- Positive: 1,4470096
- Negative: -6,5550632

Top positive words:

1. poignant: 0.4932554957031941
2. great: 0.2576831567161181
3. impressive: 0.1924303382745759

Top negative words:

1. formulaic: -0.4878360219732369
2. poor: -0.3495095422162071
3. bad: -0.17247701342408656

The top influencing words are exactly the same as in the unchanged model for this review. The influence values of the negative words are all slightly lower, which leads to an even more definite classification.

Positive review 2 (score 9 of 10)

Classified as positive by the neural network with the scores:

- Positive: -1.4660962
- Negative: -2.6034153

Top positive words:

1. favorite: 0.3237694061155647
2. entertaining: 0.2097175098063043
3. quite: 0.14302447415518868

Top negative words:

1. awful: -0.7723869133619864
2. joke: -0.17855818001522294
3. n't: -0.11637205339274101

This review was classified correctly, where it was not classified correctly by the unchanged model. The top words and their influence values only differ slightly.

Negative review 1 (score 3 of 10)

Classified as negative by the neural network with the scores:

- Positive: -2.6650193
- Negative: -2.543482

Top positive words:

1. thoughtful: 0.5863735813957756
2. superior: 0.1840688999840179
3. years: 0.15087119408285066

Top negative words:

1. blood: -0.24636816251202426
2. shame: -0.21988787374399257
3. execution: -0.20653802202989088

This review is classified correctly, but for the wrong reasons. The words "blood" and "shame" can relate to a negative sentiment, but for movies this kind of words describe the content of the movie and not the opinion about it. Again the word "thoughtful" was influencing the classification although it is not meant in the context of the movie but another review.

Negative review 2 (score 2 of 10)

Classified as positive by the neural network with the scores:

- Positive: -0.7083007
- Negative: -4.685843

Top positive words:

1. superbly: 0.40959423217730095
2. great: 0.38584245487589564
3. impressive: 0.25920779294161356

Top negative words:

1. lacked: -0.36029218475306213
2. cinematography: -0.32170126387905174
3. nothing: -0.2541318055516034

There is not much difference for this review compared to the unchanged model result. This model still deals with the same problems.

Summary

Compared to the unchanged model, this model performed quite similar. Surprisingly it classified the negative review 2 correctly where most of the other models failed. By the results one cannot really tell which model performed better regarding to capture the sentiment of those reviews.

4.4.2 Decrease number of filters

- Filter set sizes: 3, 4 and 5
- Number of filters per filter set: 32
- Use pre-embedded data: True
- Embedding size: 300
- Max number of words: 300
- Number of training epochs: 10
- Training batch size: 128

The trained network has an accuracy of 87.98% on the evaluation data which is slightly lower than the accuracy of 88.62% of the unchanged model.

Positive review 1 (score 9 of 10)

Classified as positive by the neural network with the scores:

- Positive: 1,9482627
- Negative: -3,189998

Top positive words:

1. poignant: 0.6530830766731296
2. drama: 0.2924730952589021
3. great: 0.22118614307852064

Top negative words:

1. formulaic: -0.4128394206495033
2. poor: -0.20208460841150858
3. director: -0.13449801235391093

This review was classified correctly. The word "drama" did not occur in the unchanged mode and it is not highlighted at all in the visual output. This model put a big positive value into this word. The word "bad" is not appearing here, instead there is the word "director" which also was a top negative word of other results.

Positive review 2 (score 9 of 10)

Classified as negative by the neural network with the scores:

- Positive: -1.1836097
- Negative: -0.29381624

Top positive words:

1. favorite: 0.37838953703794165
2. entertaining: 0.35381082488106513
3. enjoy: 0.21534677383561726

Top negative words:

1. awful: -0.6555505808022287
2. muster: -0.17083222859796462
3. an: -0.1269673688573718

This review was classified correctly, where it was not classified correctly by the unchanged model. The top positive words are the same and again "awful" has a high negative influence, but the influence of "awful" in this classification is lower than in the unchanged model.

Negative review 1 (score 3 of 10)

Classified as positive by the neural network with the scores:

- Positive: -0.35967517
- Negative: -0.72366315

Top positive words:

1. thoughtful: 0.6710196459877398
2. superior: 0.22767223852412247
3. vulnerability: 0.15273038973252073

Top negative words:

1. wrong: -0.2584791176266219
2. director: -0.18247474657996604
3. n't: -0.17529871520287105

The top words for this review are close to the top words from the unchanged model. It is also classified incorrectly. The word "vulnerability" is appearing here as a quite positive word, where there was the word "interesting" for the unchanged model.

Negative review 2 (score 2 of 10)

Classified as positive by the neural network with the scores:

- Positive: -0.804853
- Negative: -2.2600012

Top positive words:

1. superbly: 0.5517800520961815
2. impressive: 0.3959852227029842
3. job: 0.19559212812255636

Top negative words:

1. lacked: -0.42899098744194697
2. nothing: -0.21362220225656217
3. movie: -0.15161087704491272

This review is classified incorrectly, because of the high influences of the few positive words in this review. In most other approaches the word "cinematography" was influencing the negative classification heavily, but it does not appear for this model.

Summary

This model showed no major flaws. It just has an slightly smaller accuracy on the whole evaluation data and a few top influencing words differed from the ones of the unchanged model, but most of them are still sentimental.

4.5 Change training parameters

Another way to influence the training is to change the training parameters. As described in 2.4, the training parameters I considered were the batch size and the number of epochs. I chose to change the number of epochs, because it changes the number of times the training data is looked at, which generally should have an effect on over fitting or under fitting.

4.5.1 Increase number of epochs

- Filter set sizes: 3, 4 and 5
- Number of filters per filter set: 128
- Use pre-embedded data: True
- Embedding size: 300
- Max number of words: 300
- Number of training epochs: 20
- Training batch size: 128

The trained network has an accuracy of 88.58% on the evaluation data which is insignificantly lower than the accuracy of 88.62% of the unchanged model.

Positive review 1 (score 9 of 10)

Classified as positive by the neural network with the scores:

- Positive: 8.445078
- Negative: -1.2960407

Top positive words:

1. great: 0.38332481722676826
2. poignant: 0.3812821775147118
3. job: 0.22231856683924842

Top negative words:

1. poor: -0.5336581570446424
2. formulaic: -0.36962872779463035
3. bad: -0.10629591155848837

The top influencing words are the same as in the unchanged model, but the order is different. The word "great" is more positive than the word "poignant" and the word "poor" is more negative than the word "formulaic" for this review.

Positive review 2 (score 9 of 10)

Classified as positive by the neural network with the scores:

- Positive: 3.825588
- Negative: 3.4267907

Top positive words:

1. entertaining: 0.3170106407116087
2. favorite: 0.2611770036498712
3. quite: 0.1281377929896604

Top negative words:

1. awful: -0.776578949085933
2. an: -0.1358300573721563
3. n't: -0.11991540904400085

The network was very indecisive on this classification, but classified the review correctly, where the unchanged model failed. This model has the same problem as the unchanged model, namely the negative influencing word "awful".

Negative review 1 (score 3 of 10)

Classified as positive by the neural network with the scores:

- Positive: 2.997063
- Negative: 2.3647919

Top positive words:

1. thoughtful: 0.42764360933986056
2. plenty: 0.17306751358642672
3. flames: 0.1703264043258296

Top negative words:

1. puzzled: -0.299831076116401
2. wrong: -0.26723367768627776
3. director: -0.2566451861441988

These top influencing words differ from the ones of the unchanged model. The words "plenty" and "flames" seem not to be sentimental words and from their context the should

not influence the classification. The classification of this review was incorrect, but again the scores for negative and positive were close to each other.

Negative review 2 (score 2 of 10)

Classified as positive by the neural network with the scores:

- Positive: 5.8790956
- Negative: 2.2862892

Top positive words:

1. impressed: 0.3792707864998326
2. great: 0.3395532494592685
3. impressive: 0.30806210926616184

Top negative words:

1. lacked: -0.368756169571591
2. nothing: -0.2372379353170699
3. cinematography: -0.22412758587625006

This model deals with the problem of few very strong positive words in this negative review differently than the unchanged model, but still fails to classify this review correctly. The word "superbly" that was the top positive word for the unchanged model does not appear here at all. Instead the top positive word here is "impressed" which can be seen as quite negative from the context of the sentence: "{...} that was the only thing which impressed me about the movie {...}". Also the word "cinematography" is a bit less negative for this model. That goes in the right direction because the word should actually be not negative at all regarding the context of the word.

Summary

Even though the neural network of this approach saw the training data more often, it still did not perform better than the unchanged model on the evaluation data. The top influencing words had some differences to the ones from the unchanged model and made in some cases more and in some cases less sense.

4.5.2 Decrease number of epochs

- Filter set sizes: 3, 4 and 5
- Number of filters per filter set: 128
- Use pre-embedded data: True
- Embedding size: 300
- Max number of words: 300
- Number of training epochs: 5

- Training batch size: 128

The trained network has an accuracy of 87.55% on the evaluation data which is slightly lower than the accuracy of 88.62% of the unchanged model.

Positive review 1 (score 9 of 10)

Classified as positive by the neural network with the scores:

- Positive: 0.53644943
- Negative: -3.5042505

Top positive words:

1. poignant: 0.48892119071775547
2. great: 0.2568739854014521
3. drama: 0.25252465009787284

Top negative words:

1. formulaic: -0.49728662906757615
2. poor: -0.2797389498229276
3. bad: -0.20258256113317571

The top influencing words are the same as the ones from the unchanged model except one word. The word "drama" seems to be no sentimental word but it is next to the word "poignant" in the review, which increases the positive influence of "drama".

Positive review 2 (score 9 of 10)

Classified as negative by the neural network with the scores:

- Positive: -1.9575589
- Negative: -1.798511

Top positive words:

1. entertaining: 0.28820371101272974
2. favorite: 0.2745459655232707
3. enjoy: 0.1585687885337094

Top negative words:

1. awful: -0.802604358166357
2. an: -0.15070879630490824
3. joke: -0.1240456005661843

This review is classified incorrectly with a very small distance between the scores of the output classes. The top influencing words are exactly the same as the ones from the unchanged model with just minor differences in their influence values.

Negative review 1 (score 3 of 10)

Classified as positive by the neural network with the scores:

- Positive: -0.9600364
- Negative: -1.5599743

Top positive words:

1. thoughtful: 0.38497073862656495
2. beholder: 0.26133826322198767
3. vulnerability: 0.22525673736946603

Top negative words:

1. wrong: -0.47357141014937426
2. director: -0.25244316870446143
3. orgy: -0.19739923060272033

This neural network has the same problem as the others finding the words in this review, that really influence the class. None of the words from the first sentence of the review are in the top influencing words, but only the first sentence really can identify the sentiment of that review.

Negative review 2 (score 2 of 10)

Classified as positive by the neural network with the scores:

- Positive: -0.09693551
- Negative: -2.5718827

Top positive words:

1. impressive: 0.44959873743071405
2. superbly: 0.44532054449590075
3. great: 0.3893600696621323

Top negative words:

1. lacked: -0.2880978448707983
2. cinematography: -0.24387665021045882
3. movie: -0.12126358531140143

Like most of the other models failed at this review, this neural network also attributed the few positive words with a very high influence on the classification. So in the end the negative words cannot turn the result around.

Summary

This approach lead to similar results than the unchanged model, despite seeing the training data only half as often while training. Most of the top influencing words had a clear sentiment. Due to its slightly lower accuracy, this approach seems to be inferior to the unchanged model.

4.6 Summary

The table 4.1 shows an overview on the accuracy and the classification of the example evaluation reviews for each of the experiments. The best result is achieved by adding the 4 evaluation reviews to the training data but all of them are very close to each other regarding their accuracy. The achieved accuracies are also slightly worse than the accuracy of 88.89% achieved by the creators of the data set. In the following chapter

Model	Accuracy on the whole data set	Pos. 1 class.	Pos. 2 class.	Neg. 1 class.	Neg. 2 class.
Unchanged	88.62%	pos.	neg.	pos.	pos.
Added training data	88.69%	pos.	pos.	neg.	neg.
Trained embedding	87.38%	pos.	neg.	pos.	neg.
More filters	88.62%	pos.	neg.	neg.	pos.
Less filters	87.98%	pos.	neg.	pos.	pos.
More epochs	88.58%	pos.	pos.	pos.	pos.
Less epochs	87.55%	pos.	neg.	pos.	pos.

Table 4.1: Experiment results

the results of the experiments and how the changes influenced the actual quality will be discussed.

5 Improving classification

The goal of the experiments were to find general suggestions how to change a text classification neural network to improve its classification quality. Due to the small scale of the experiments, the results of all experiments are quite similar. All of the results have to be validated. Using the same approach this can easily be done for more different and bigger changes to the neural network and its training. Still some consequences can be taking out of the results of the experiments.

Adding additional training data

This approach seemed to be very good regarding the general result. All of the example evaluation reviews were classified correctly and even the accuracy on the whole evaluation data was slightly higher. The problem is that those reviews are corner cases that were learned by the neural network. The neural network learned some words to be deciding that generally have no sentiment. Because there is no counter example in the training data, those corner cases are memorized. Generally one wants the neural network to learn rules. With adding just a few more data points to the training data, that were classified wrongly before, it is hard for the neural network to extract the rules from them. A solution to the problem would be to add a bunch of additional training data. This way either there are multiple data points with such a corner case and the network manages to find a rule in them or it will vanish as noisy data. Such corner cases can only vanish if the neural network is small enough to not be over fitting.

Train the word embedding

The experiment showed that on the training data set of 25000 reviews, the neural network did not manage to learn all the words that are in the 25000 evaluation data set. Compared to the neural network that used the pre-embedded model, most of the highlighted words were less sentimental and some of the very sentimental words were not highlighted at all. In order to train a proper word embedding, all the words in the evaluation data that are important for the classification have to be present in the training data. In some cases it is beneficial to train the word embedding, namely cases were the classification depend on a few very influencing words, that need to be in all the data points. That way the training of the embedding can separate the embedding vectors of those highly influencing words to better distinguish them. When dealing with smaller data sets and having a common understanding or sentiment problem, one should better use a pre-embedded model.

Change number of filters and training epochs

There is no general rule visible regarding results of the experiments that changed the number of filters and the number of training epochs. All of the results had a similar accuracy on the whole data set and all of them had sentimental words in their list of

highest influencing words. Of course some changes will definitely change the quality of the neural network but the changes of the experiment were not big enough to have a heavy impact.

6 Conclusion

In this thesis, I gave an approach on how to applying the approach of "What does my Classifier do?"[1] to get a better understanding of the neural networks and find ways to improve them. Different neural networks can be compared apart from just looking at the accuracy. This lead to a good understanding of how additional training examples and trained word embedding changed the training of a neural network and its quality. However the results of the other experiments were not so convincing. In order to really understand what other changes do with the neural network further experiments are required. Still to find the best parameters and architecture for a neural network dealing with a certain problem one need to test different approaches but the comparison of the different solution is more expressive using this approach. The experiments described in this thesis seem to be not heavy enough in order to really change the resulting neural network. Also this approach could be applied to other domains, apart from sentiment analysis, using more and heavier changes to the neural network. Staying in the field of natural language processing, where the input and output is easily understandable, this approach gives additional ways to compare neural networks and their interior. Additionally a further future work could be to apply this approach to other domains like image recognition. Also it would be interesting to extend this approach to not be only applicable to convolutional neural networks but also other neural network types like recurrent neural networks.

Bibliography

[1] Jonas Paul Winkler, Andreas Vogelsang: *"What does my classifier learn?" A visual approach to understanding natural language text classifiers*, TU Berlin, 2018

[2] Andrew L. Maas, Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, Christopher Potts: *Learning Word Vectors for Sentiment Analysis*, Stanford University, 2011

[3] Yoav Goldberg: *A Primer on Neural Network Models for Natural Language Processing*, Stanford University, 2015

[4] Tomas Mikolov, Kai Chen, Greg Corrado, Jeffrey Dean: *Efficient Estimation of Word Representations in Vector Space*, Google Inc., Mountain View, CA, 2013

Evaluated review examples

1 Positive

1. Rating 9 of 10: "A talented high school graduating senior with a bad attitude is forced to play in the state all-star high school football game. When he meets and falls for an attractive local girl she helps him realize he has a shot at a 'full ride' scholarship if he plays well.

All too often, these dramas fall into formulaic traps and tell the same old story of a troubled and confused teen. FULL RIDE's Matt Sabo certainly fits this profile, but below the surface is a much more unique individual than we usually see in this genre. Matt is the center of the action and he is a realistic teenager, both over-confident and vulnerable, optimistic and cynical by turns. Influenced by Amy, Matt grows into a man of character and heart. He, in turn, forms friendships with his teammates, which influences his growth as an athlete and as a team player.

FULL RIDE has all the elements we love to see in a movie–great acting, admirable characters, exciting sports scenes, poignant drama, and a love story. Still, while one may have seen these elements in other films, FULL RIDE is assisted by performances that are sincere and occasionally, even moving. Perhaps what's most impressive about FULL RIDE is its sense of reality. Although the author of the previous comment would seem to disagree, (clearly a disgruntled student who, for quite obvious reasons, received a poor grade in his film class) director Mark Hoeger grounds the film in a believable situation and location and does a great job of getting down to the grit of what life is like in a small town. These characters are real people rooted in realistic situations, which often create the most compelling entertainment. On one level it is a love story, on another it is a character study, and yet another it is a simple football film. All of these ideas come together to form a cohesive vehicle."

2. Rating 8 of 10: "If you keep rigid historical perspective out of it, this film is actually quite entertaining. It's got action, adventure and romance, and one of the premiere casting match-ups of the era with Errol Flynn and Olivia de Havilland in the lead roles. As evident on this board, the picture doesn't pass muster with purists who look for one hundred percent accuracy in their story telling. To get beyond that, one need only put aside the history book, and enjoy the story as if it were a work of fiction. I know, I know, that's hard to do when you consider Custer's Last Stand at the Little Big Horn and it's prominence in the history of post Civil War America. So I guess there's an unresolved quandary with the picture, no matter how you look at it.

There's a lot to take in here though for the picture's two hour plus run time. Custer's arrival at West Point is probably the first head scratcher, riding up as he does in full military regalia. The practical joke by Sharp (Arthur Kennedy) putting him up in the Major's headquarters probably should have gotten them both in trouble.

Ironically, a lot of scenes in

this military film play for comedy, as in Custer's first meeting with Libby Bacon, and subsequent encounters that include tea reader Callie (Hattie McDaniel). I hadn't noticed it before in other films, but McDaniel reminded me an awful lot of another favorite character actor of mine from the Forties, Mantan Moreland. So much so that in one scene it looked like it might have been Moreland hamming it up in a dress. With that in mind, the owl scene was a hoot too.

As for Flynn, it's interesting to note that a year earlier, he portrayed J.E.B. Stuart opposite Ronald Reagan's depiction of General Custer in "Santa Fe Trail", both vying for the attention of none other than Olivia de Havilland. In that film, Reagan put none of the arrogance and flamboyance into the character of Custer that history remembers, while in Flynn's portrayal here it's more than evident. But it doesn't come close to that of Richard Mulligan's take on the military hero in 1970's "Little Big Man". Let's just say that one was a bit over the top.

The better take away the picture had for me was the manner in which Custer persevered to maintain his good name and not gamble it away on a risky business venture. That and his loyalty to the men he led in battle along with the discipline he developed over the course of the story. Most poignant was that final confrontation with arch rival Sharp just before riding into the Little Big Horn, in which he declared that hell or glory was entirely dependent on one's point of view. Earlier, a similar remark might have given us the best insight of all into Custer's character, when he stated - "You take glory with you when it's your time to go"."

2 Negative

1. Rating 3 of 10: "While it's true that the movie is somewhat interesting, the execution leaves a lot to be desired (much like Blood Orgy of the Leather Girls, I Spit on Your Grave, and Born in Flames, all superior). I don't think it's not porn, but porn is in the eye of the beholder: if it functions as porn for somebody, who am I to say that he/she is wrong? I was rather puzzled by the statement in Winkimation's generally thoughtful review ("Such a Shame") that "for once we actually see men's faces when they come." A few years ago I did occasional freelance reviews for an adult mag and I recall seeing plenty of men's faces when they came. I think this is probably more common when the film features on of the few male porn "stars" (and especially when that male is the director). Though I unsurprisingly can't refer to any specific titles, I know that there are some instances in Ron Jeremy's, uh, work. I also don't know that I'd agree that a man is necessarily showing vulnerability in his face when coming."

2. Rating 2 of 10: "Water shows the plight of Indian widows in the late 1930s, says in the end that the problem still exists largely by giving statistics in the end, refers to Gandhi several times in the movie before finally having a scene depicting him and does nothing extra ordinarily innovative or new in the movie. Yes, the cinematography is pretty impressive but that cannot be the soul of any movie for me.

India has had several problems like many other nations but

it has got rid of many of these problems at large. What if a movie is made on racism in America in a particular year which ends with 'x number of Americans still experience racism today'.

a) How would it be relevant, and, b) How would it be some thing so extra ordinary being depicted in cinema.

A view I read from a Deepa Mehta interview was that this movie is being interpreted as a voice for the marginalised every where. From reviews I read every where, the common thing I am hearing is how the director did a great job and was brave in bringing a problem to the world. The movie is more about a specific problem a society faced (and has got rid of through reforms at large).

I do not see any thing earth shattering about the movie. Moreover, the movie lacked soul and shifted between the plots of Chuiyya and Kalyani. Sarala, the young Sri Lankan actress, portrayed the role of Chuiyya superbly and that was the only thing which impressed me about the movie, sadly."