Machine Learning Engineer Nanodegree

Capstone Proposal

Rohan Asnani
May 5th, 2020

Efficient detection of deceptive content using Deep Learning

Domain Background

What is deceptive content and why is it so important to detect it?

The usage of social media is increasing at a very high rate more and more users are using social networks for different reasons. In 2010 there were 0.97 billion users on social media and currently, 3.03 billion users are there and it is estimated to touch 3.43 billion till 2023[1]. These users use the social media platform for various reasons one of them is spreading awareness and news among the other users on the same platform.
There are many times that people spread false or fake news intentionally the reasons why people or organizations spread fake news can be to increase their popularity, for political reasons i.e to spread fake news about the opposing party so that the opposing party does not receive enough votes. In 2016 U.S.A elections there was a large amount of fake news on Twitter and Facebook which were in favour of Donald Trump and false rumours about Clinton's party, it was found that pro-Trump fake stories that were shared were around 30 million times and pro-Clinton fake stories shared were around 7.6 million times[2], In vengeance and many other reasons.
Recently there were many screenshots and tweets being forwarded that a man in Kerala fed a pineapple filled with fire-crackers to a pregnant elephant which resulted in the death of the elephant, which went viral and created a sad and emotional atmosphere among the people. The horrific death of an elephant in Kerala's Palakkad was yet another issue to target the Indian Muslim population. After reports came pouring in that the animal died a slow death after explosives went off in her mouth, numerous social media users claimed that the persons arrested for the crime are Amzath Ali and Tamim Shaikh. Amar Prasad Reddy, media advisor to the Union Minister of State for Health and Family Welfare, was one of the earliest to tweet. He later took it down but not before it garnered thousands of likes and retweets[3]. There are many examples of such news which created huge havoc among the people, The mainstream media has been the biggest contributor of fake news which has become a big issue now even the prime minister of India Mr Narendra Modi addressed the nation about how fake news has become a very big problem and we shouldn't believe anything forwarded over the social media platforms.

Therefore there is an urgent need to detect and eliminate fake news immediately.

Detection of fake news

With the increase in the number of users online and the advancement of technology, the amount of digital news is exposed to users globally and this contributes to the increase of spreading of fraud and misinformation online. Fake news can be easily found online over the social media platform.

Fake news can be detected using neural networks that analyze the texts and marks them as fake, real, rumour, or debatable.

Types of fake news

1) NEWS CONTENT BASED:-

   This is also known as a visual-based type of fake news that includes photo-shopped images, visual news, the spread of historical-based pictures attached to a recent one to create belief, etc.

   Facebook, twitter, telegram, WhatsApp, Instagram, etc. are some of the examples of social media platforms where such content-based fake news is shared

2) NEWS CONTEXT BASED:-

   This is also known as Linguistics based type of fake news this news is in the form of strings or texts.

   Gmail, twitter, Whatsapp, blog sites, etc., are some examples of social media platforms where such context-based images are shared

3) NEWS TEMPORAL BASED:-

   In this type, the news can be in content or context-based which is posted at first and after creating a huge hoax is deleted or removed later making it hard to find the source.

   Any social media platform can be used to spread such news.

4) NEWS CREDIBILITY BASED:-

   This means analyzing public fake news data sources and authors that can be the indicator if the news is credible or not by analyzing the history of the author and the source of information.

   This type of detection can play a major role in improving the efficiency of detecting fake news.

Problem Statement

Deceptive content is potentially harmful and can take many forms
The proliferation of digital fake news is one of the most pressing needs of modern society, with several academics as well as tech companies investing heavy resources for this task.
We categorize this as an Artificial Intelligence problem, and suggest Deep Learning methods to tackle it. The problem is viewed as the classification of a single news story based on the features of the online posts users make about the news, using Twitter15 and Twitter16 datasets to show the same.

Current Research and Methodology Trends

Detection Strategy	Data	Method	Model
Knowledge-based	Features, tags and keywords extracted from the content.	*Knowledge extraction* to collect raw facts from multiple credible sources on the internet. Represented in entityrelationship graph, which is used to verify the suspected content.	Unsupervised RNN and supervised LSTMs with frameworks like Keras and concepts like POS (parts-of-speech tagging).
Content-based	Entities and relationships extracted from the text.	Language feature extraction from the text based on attributes like complexity of language, the structure of the paragraphs, ambiguity and contradiction, semantics, syntax, and so on.	Supervised machine learning classification or regression using models such as Naive Bayes, kNN, Singular Vector Machines, RIPPER.
Propagation-based	Trees/Forests, Graphs structure of the content passing from source to all subsequent interacting users.	Represent sequence of sources and users as nodes in graph and train model to detect suspicious propagation.	LSTM, RNN, and/or CNN for sequence of propagation over time, for graphs: SEIZ Contagion Model, GBDT (Gradient Boosting with Decision Trees), GNN (Graph

			Neural Networks)
Credibility-based	Can use content, behavior (temporal), user interaction graph.	Map sources to their credibility defining attributes - like page hits, ranks, citation in other reliable sources, sentiments about sources, to classify whether a source historically or currently delivers factually correct information.	Ranking algorithm like PageRank, Logistic regression and Linear Support Vector Machines for classification using selected attributes.

Datasets and input

Datasets

1. Fake News [1]
2. Fake or Real News [2]
3. Election Day Tweets [3]
4. Fake News Data [4]
5. LIAR
6. TWITTER15&16

Fake News Dataset	
Columns/Attributes	1. ID 2. Title 3. Author 4. Text 5. Label (0, 1)
Rows	20,800
Type	Content-based
Attributes considered	Title, Text, Label
Split ratio (train/val/test)	16640 10400 10400

Election Day Tweets	
Columns/Attributes	1. ID 2. Text 3. Retweet count 4. Username 5. (12 more)
Rows	1327
Type	Content-based
Attributes considered	text, label, retweet
Split ratio (train/val/test)	1061 664 663

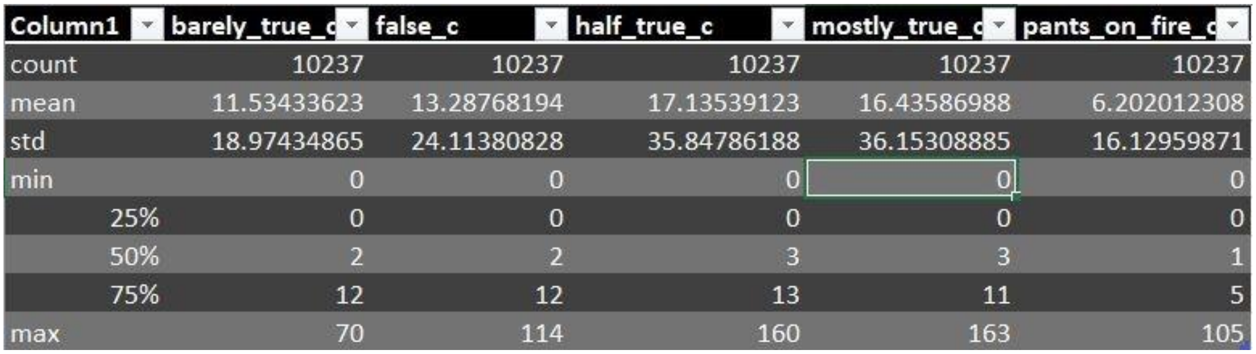
Colu
Row
Type
Attri
Split (trai

Colu
Row
Type
Attri
Split (trai

LIAR

SPLIT	TRAIN	VALIDATION	TEST
SIZE	9727	6144	4095

FALSE	1892	1218	776
TRUE	1582	1008	668
MOSTLY-TRUE	1870	1142	820
HALF-TRUE	2000	1261	853
PANTS ON FIRE	798	503	336
BARELY TRUE	1585	1012	642



Column1	barely_true_c	false_c	half_true_c	mostly_true_c	pants_on_fire_c
count	10237	10237	10237	10237	10237
mean	11.53433623	13.28768194	17.13539123	16.43586988	6.202012308
std	18.97434865	24.11380828	35.84786188	36.15308885	16.12959871
min	0	0	0	0	0
25%	0	0	0	0	0
50%	2	2	3	3	1
75%	12	12	13	11	5
max	70	114	160	163	105

DESCRIPTION OF THE LIARDATASET

TWITTER 15&16

For experimental evaluation, we use two publicly available Twitter datasets released by Ma et al. (2017), namely Twitter15 and Twitter164 , which respectively contains 1,381 and 1,181 propagation trees (see (Ma et al., 2017) for detailed statistics). In each dataset, a group of wide spread source tweets along with their propagation threads, i.e., replies and retweets, are provided in the form of tree structure. Each tree is annotated with one of the four class labels, i.e., non-rumor, false rumor, true rumor and unverified rumor. We remove the retweets from the trees since they do not provide any extra information or evidence contentwise. We build two versions for each tree, one for the bottom-up tree and the other for the top-down tree, by flipping the edges' direction.

Model Architectures

Introduction

Neural Network model architectures have recurring structures based on the type of neural network: such as convolutional, recurrent, etc. In this section the types of common layers, their purposes and functions are briefly discussed.

CNN[5]

A Convolutional Neural Network is a deep learning algorithm that can recognize and classify features in data such as images or text. It is a multi-layer (deep when layers are dense) neural network designed to analyze visual inputs and perform tasks such as image classification, segmentation, object detection, or text classification, which are useful in various countless scenarios.[6]

A CNN is composed of several kinds of layers:

- **Convolutional layer:** creates a feature map to predict the class probabilities for each feature by applying a filter that scans the whole image, few pixels at a time.
- **Pooling layer (downsampling):** scales down the amount of information the convolutional layer generated for each feature and maintains the most essential information (the process of the convolutional and pooling layers usually repeats several times).
- **Fully connected input layer:** "flattens" the outputs generated by previous layers to turn them into a single vector that can be used as an input for the next layer.
- **Fully connected layer:** applies weights over the input generated by the feature analysis to predict an accurate label.

- **Fully connected output layer:** generates the final probabilities to determine a class for the image.

LSTM[7]

The German researchers, Hochreiter and Schmidhuber, introduced the idea of long short-term memory networks in a paper published in 1997. LSTM is a unique type of Recurrent Neural Network (RNN) capable of learning long-term dependencies, which is useful for certain types of prediction that require the network to retain information over longer time periods, a task that traditional RNNs struggle with.[8]

The chain-like architecture of LSTM allows it to contain information for longer time periods, solving challenging tasks that traditional RNNs struggle to or simply cannot solve. The three major parts of the LSTM include:

- **Forget gate**—removes information that is no longer necessary for the completion of the task. This step is essential to optimizing the performance of the network.

- **Input gate**—responsible for adding information to the cells

- **Output gate**—selects and outputs necessary information

Other Key Terms

Batch Normalization: method to standardize inputs going into a network, in order to proceed the activations of a Preceding layer and its inputs directly.

Regularization: This is a form of regression, that constrains/ regularizes or shrinks the coefficient estimates towards zero. In other words, this technique discourages learning a more complex or flexible model, so as to avoid the risk of overfitting.[9]

Model Architectures Used for Given Datasets

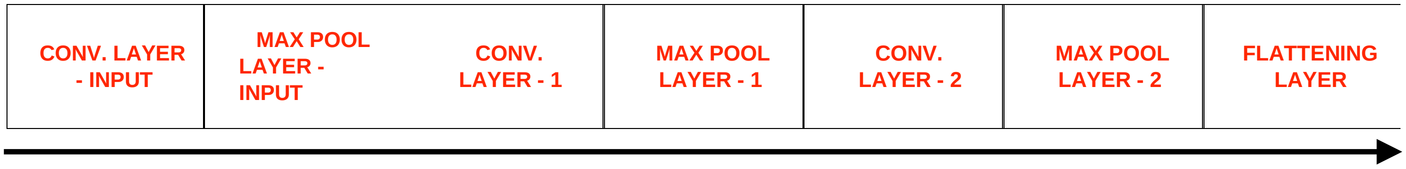
CNN Model

```
for fsz in filter_sizes:
l_conv =
Conv1D(nb_filter=128,filter_length=fsz,activation='relu')(embed-
ded_sequences)
    l_pool = MaxPooling1D(5)(l_conv)
    convs.append(l_pool)

l_merge = concatenate(convs, axis=1)
l_cov1= Conv1D(filters=128, kernel_size=5, activation='relu')
(l_merge)
l_pool1 = MaxPooling1D(5)(l_cov1)
l_cov2 = Conv1D(filters=128, kernel_size=5, activation='relu')
(l_pool1)
l_pool2 = MaxPooling1D(30)(l_cov2) l_flat
= Flatten()(l_pool2)
l_dense = Dense(128, activation='relu')(l_flat) preds
= Dense(2, activation='softmax')(l_dense)

model2 = Model(sequence_input, preds)
model2.compile(loss='categorical_crossentropy',
              optimizer='adadelta',
```

As we can see in this code snippet where the CNN model is being designed, we have the following layers in the CNN architecture for Fake News Dataset:



CONV. LAYER - INPUT

MAX POOL LAYER - INPUT

CONV. LAYER - 1

MAX POOL LAYER - 1

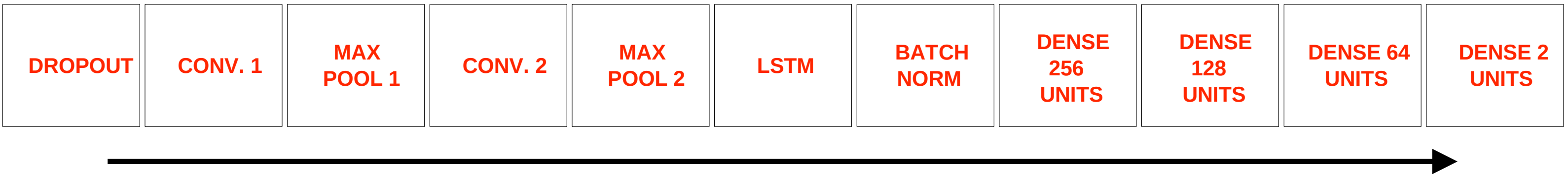
CONV. LAYER - 2

MAX POOL LAYER - 2

FLATTENING LAYER

Activation Function: Input => ReLU
Output => softmax
Loss Function: Categorical Cross Entropy
Optimizer: Adadelta

LSTM Model



DROPOUT

CONV. 1

MAX POOL 1

CONV. 2

MAX POOL 2

LSTM

BATCH NORM

DENSE 256 UNITS

DENSE 128 UNITS

DENSE 64 UNITS

DENSE 2 UNITS

```
embedding_vecor_length = 32 modell
= Sequential()
modell.add(embedding_layer)
modell.add(Dropout(0.2))
modell.add(Conv1D(filters=32, kernel_size=5, padding='same', activation='relu'))
modell.add(MaxPooling1D(pool_size=2))
modell.add(Conv1D(filters=64, kernel_size=3, padding='same', activation='relu'))
modell.add(MaxPooling1D(pool_size=2))
modell.add(LSTM(100, dropout=0.2, recurrent_dropout=0.2))
modell.add(BatchNormalization())
modell.add(Dense(256, activation='relu'))
modell.add(Dense(128, activation='relu'))
modell.add(Dense(64, activation='relu'))
modell.add(Dense(2, activation='softmax'))
modell.compile(loss='categorical_crossentropy', optimizer='adam',
metrics=['accuracy'])
```

As we can see in this code snippet where the LSTM model is being designed, we have the following layers in the LSTM architecture for Fake News Dataset: AFTER AN INITIAL *EMBEDDING LAYER*

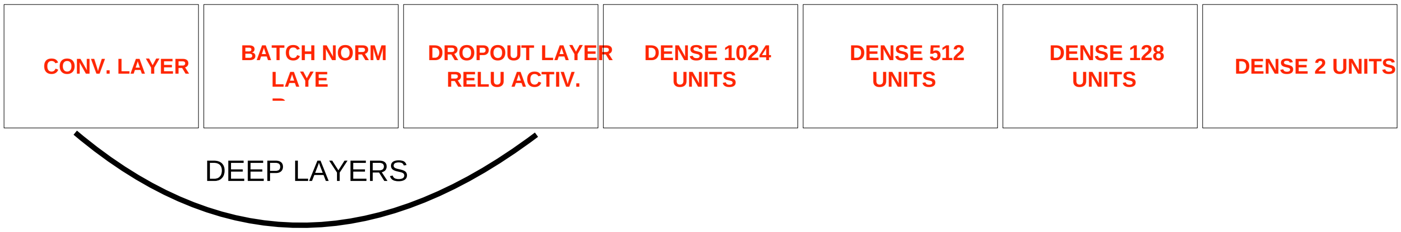
Activation Function: Input => ReLU
Output => softmax
Loss Function: Categorical Cross Entropy
Optimizer: Adam

DeepCNN

```
modell.add(embedding_layer) for i in range(0,2):     modell.add(Conv1D(filters=1024,
kernel_size=1, padding='same', activation='relu'))
modell.add(BatchNormalization())

for i in range(0,5):     modell.add(Conv1D(filters=32, kernel_size=5,
padding='same', activation='relu'))
    modell.add(BatchNormalization())
modell.add(Activation('relu'))
     for i in range(0,5):     modell.add(Conv1D(filters=64, kernel_size=3,
padding='same', activation='relu'))
    modell.add(BatchNormalization())
modell.add(Activation('relu'))
     for i in range(0,3):     modell.add(Conv1D(filters=64, kernel_size=5,
padding='same', activation='relu'))
    modell.add(BatchNormalization())
modell.add(Activation('relu'))
    modell.add(Conv1D(filters=128, kernel_size=3, padding='same', activation='relu'))
    modell.add(BatchNormalization())
modell.add(MaxPooling1D(pool_size=2))
modell.add(Activation('relu'))
     for i in range (0,7):     modell.add(Conv1D(filters=128, kernel_size=5,
padding='same', activation='relu'))
    modell.add(BatchNormalization())     modell.add(Activation('relu')) for i in
range (0,5):     modell.add(Conv1D(filters=256, kernel_size=3, padding='same',
activation='relu'))
    modell.add(BatchNormalization())
modell.add(Activation('relu'))
     for i in range (0,3):     modell.add(Conv1D(filters=256, kernel_size=5,
padding='same', activation='relu'))
    modell.add(BatchNormalization())
     for i in range (0,5):     modell.add(Conv1D(filters=512, kernel_size=3,
padding='same', activation='relu'))
    modell.add(BatchNormalization())
    modell.add(Dropout(0.1))
     for i in range(0,2):     modell.add(Conv1D(filters=768, kernel_size=5,
padding='same', activation='relu'))
    modell.add(BatchNormalization())     modell.add(MaxPooling1D(pool_size=2))
modell.add(Activation('relu')) for i in range(0,2):
modell.add(Conv1D(filters=1024, kernel_size=3, padding='same', activation='relu'))
    modell.add(BatchNormalization())
modell.add(Activation('relu'))

modell.add(Dense(1024, activation='relu'))
modell.add(Dense(512, activation='relu'))
modell.add(Dense(128, activation='relu'))
modell.add(Dense(2, activation='softmax'))
modell.compile(loss='binary_crossentropy', optimizer='adam', metrics=['accuracy'])
```



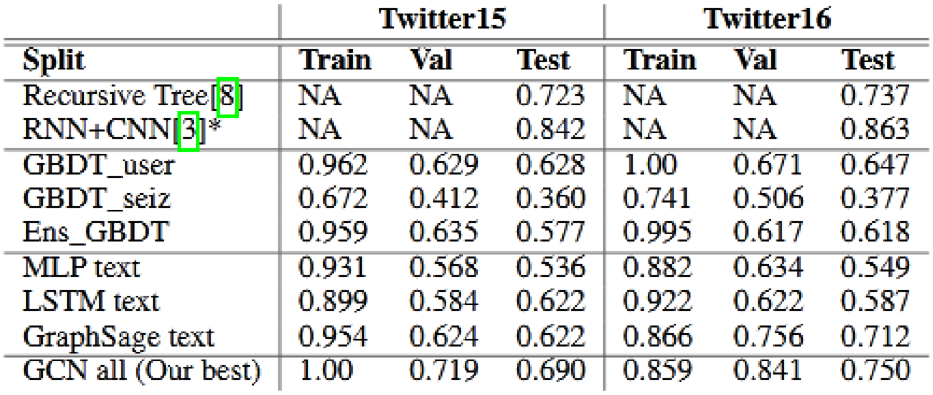
Activation Function: Input => ReLU
Output => softmax
Loss Function: Binary Cross Entropy
Optimizer: adam

Challenge: - Develop an efficient deep neural network for detection of deceptive content

Dataset: - Twitter15 and Twitter16 dataset

The datasets we work with are Twitter15 and Twitter16. These two datasets share the same exact structure. Both of them contain the tweets and re-tweets from a thousand of news articles published in 2015 and 2016. For each news article, the data contains the first tweet that shared it on Twitter, and a sequence of re-tweets following this initial post. We show one such data point (initial tweet and first two re-tweets). Each event is labeled according to the initial news article, the label is taken out of four possible classes: "true", "false", "unverified", "non-rumor". Labels are evenly distributed in both datasets**.**

Previous benchmark Results: - Previous benchmark results (Till Now) are shown below and accuracy is not so high (approx. 69%) using twitter15 dataset.



	Twitter15			Twitter16		
Split	Train	Val	Test	Train	Val	Test
Recursive Tree[8]	NA	NA	0.723	NA	NA	0.737
RNN+CNN[3]*	NA	NA	0.842	NA	NA	0.863
GBDT_user	0.962	0.629	0.628	1.00	0.671	0.647
GBDT_seiz	0.672	0.412	0.360	0.741	0.506	0.377
Ens_GBDT	0.959	0.635	0.577	0.995	0.617	0.618
MLP text	0.931	0.568	0.536	0.882	0.634	0.549
LSTM text	0.899	0.584	0.622	0.922	0.622	0.587
GraphSage text	0.954	0.624	0.622	0.866	0.756	0.712
GCN all (Our best)	1.00	0.719	0.690	0.859	0.841	0.750

my Goal: -To apply deep learning models using ALL THE datasets for fake news Detection that are mentioned.

Future Research and Conclusion

Several sub-problems and pre-problems exist in the area of fake news detection, one such problem being to detect whether a post is important or dangerous enough to employ fake news detection. It is also a challenge to detect fake news prior to its propagation, when serious effects of it have little chance of having already taken place.

Deep Learning is greatly advantageous in improving over present performance levels. Application of improved and customized CNN, Deep CNN, LSTM, and RNN models to the different detection methods can yield constantly improving results as datasets continue to improve.

Today, technology and social media companies are realizing the importance of the quality of information their users are posting and receiving. And they are slowly beginning to prioritize this over maximizing attention of users in their platforms. Seeing as attention has been the highest currency on the internet until now, this is a welcome change.

REFERENCES:-

[1]https://www.statista.com/statistics/278414/number-of-worldwide-social-network-users/
[2]https://web.stanford.edu/~gentzkow/research/fakenews.pdf
[3]https://thewire.in/communalism/kerala-elephant-death-fake-news

1. https://www.kaggle.com/c/fake-news/data
2. https://www.kaggle.com/rchitic17/real-or-fake
3. https://zenodo.org/record/1048820#.XvSmSi0w1bU
4. https://www.kaggle.com/antmarakis/fake-news-data
5. Lecun, Y., Haffner, P., Bottou, L., & Bengio, Y. (1999). Object Recognition with GradienBased Learning.Â Shape, Contour and Grouping in Computer Vision Lecture Notes in Computer Science,Â 319-345. doi:10.1007/3-540-46805-6_19
6. https://missinglink.ai/guides/convolutional-neural-networks/convolutional-neural-network-architecture-forging-pathways-future/
7. https://arxiv.org/pdf/1503.04069.pdf 8. https://missinglink.ai/guides/neural-network-concepts/deep-learning-long-short-term-memorylstm-networks-remember/ 9. https://towardsdatascience.com/regularization-in-machine-learning-76441ddcf99a

[8]O'Shea, K., & Nash, R. (2015, December 02). An Introduction to Convolutional Neural
Networks. Retrieved June 24, 2020, from https://arxiv.org/abs/1511.08458v2

[9]G. Huang, Z. Liu, L. Van Der Maaten and K. Q. Weinberger, "Densely Connected Convolutional Networks," 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, 2017, pp. 2261-2269, DOI: 10.1109/CVPR.2017.243.