Hate Speech Detection In Memes Using Multi Model Deep Learning

01CS6692 Mini Project Report

Submitted in partial fulfillment of
the requirements for the award of M.Tech Degree in
Artificial Intelligence
of the A P J Abdul kalam Technological University

Submitted by

NIZAMUDHEEN T I
KTU Reg No: TVE20CSAI13
Second Semester
M.Tech Artificial Intelligence

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
COLLEGE OF ENGINEERING
TRIVANDRUM

2021

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
COLLEGE OF ENGINEERING
TRIVANDRUM

CERTIFICATE

This is to certify that this Mini project report entitled **"Hate Speech Detection In Memes Using Multi Model Deep Learning "** is a bonafide record of the work done by **Nizamudheen T I**, under our guidance towards partial fulfilment of the requirements for the award of the Degree of **Master of Technology** in **Artificial Intelligence**, of the A P J Abdul Kalam Technological University during the year 2020-2022.

Dr. Sreeni K G
Associate Professor
Project Guide
Department of Electronics
& Communication, CET

Dr. Sumesh Divakaran
Professor
Head of Department
Department of Computer
Science & Engg., CET

Dr. Ajeesh Ramanujan
Assistant Professor
Project Coordinator
Department of Computer
Science & Engg. , CET

Prof. Rini Vijayan
Assistant Professor
Project Coordinator
Department of Computer
Science & Engg., CET

ACKNOWLEDGEMENTS

I have great pleasure in expressing my gratitude to **Dr. Jiji C.V**, the Principal, College of Engineering, Trivandrum and **Dr. Sumesh Divakaran**, Professor, Head of Department, Department of Computer Science and Engineering, College of Engineering, Trivandrum for their valuable guidance and suggestions to make this work a great success.

I express my gratitude to **Dr. Sreeni K G**, Associate Professor, Electronics and Communication Engineering, College of Engineering, Trivandrum, for all the guidance, encouragement and all the necessary help extended to me for the fulfilment of this work. I also express my gratitude to **Dr. Ajeesh Ramanujan**, Assistant Professor, Computer Science and Engineering, College of Engineering, Trivandrum and **Prof. Rini Vijayan**, Assistant Professor, Department of Computer Science and Engineering, College of Engineering, Trivandrum for all their guidance through out the fulfilment of this poject.

I also acknowledge my gratitude to other members of faculty in the Department of Computer Science and Engineering, Department of Mechanical Engineering and Department of Electronics and Communication Engineering. I also acknowledge my gratitude to my family and friends for their whole hearted cooperation and encouragement.

Nizamudheen T I

ABSTRACT

Detection of hate speech from multi modal data is major issue for the single model deep learning algorithms,especially from social media memes.Memes contains both text and image data.when detecting hate speech in meme uni model will struggle and only multi model can help.

In this project we tries to solve facebook challenge (Kiela et al.,2020),which is aims to solve detection of hate speech in memes multimodel deep learning.We used a concatenation model which concatenates text features and images feature from Text model and Image model respectively.And we applied hyper parameter tuning for getting best parameters for proposed concatenation model.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

In today's world, social media platforms play a major role in influencing people's everyday life. Though having numerous benefits, it also has the capability of shaping public opinion and religious beliefs across the world. It can be used to attack people directly or indirectly based on race, caste, immigration status, religion, ethnicity, nationality, sex, gender identity, sexual orientation, and disability or disease. Hate Speech on online social media can trigger social polarization, hateful crimes. On large platforms such as Facebook and Twitter, it becomes practically impossible for a human to monitor the source and spreading of such malicious activities, thus it is the responsibility of the machine learning and artificial intelligence research community to address and solve this problem of detecting hate speech efficiently.

In tasks such as VQA and multi-modal machine translation, it has been observed that baseline models using the language domain perform well without even exploiting the multimodal understanding and reasoning. However, the Facebook Hateful Memes Challenge dataset is designed in such a manner that unimodal models exploiting just the language or vision modalities separately will fail, and only the models that can learn the true multi-modal reasoning and understanding will be able to perform well.

They achieve this by the introduction of "benign confounders" in the dataset, i.e. for every hateful meme, they find an alternative image or caption which when replaced, is enough to make the meme harmless or non-hateful, thus flipping the label. Consider a sentence like "dishwasher for sale, missing parts". Unimodally, this sentence is harmless, but when combined with an equally harmless image of a girl without a hand, suddenly it becomes mean. Thus, this challenge set is an excellent stage that aims to facilitate the development of robust multi-modal models, and at the same time addresses an important realworld problem of detecting hateful speech on online social media platforms. Majority of the prior work baselines aim at solving this problem by finding an alignment between the two modalities, but it faces the hardship of not knowing the context behind the image and the text combination.

Chapter 2

Problem Definition and Model Design

2.1 Problem Definition

The major objective of this project is to classify memes as hateful or benign while considering their information from both text and visual modality. Denote the visual components of all memes by $X_1 = \{I_1, \ldots, I_i\}$ where i is the index of the memes, and in our case, the visual component I is the meme itself. Let $X_2 = \{T_1, \ldots, T_i\}$ denotes the text extracted from the memes. If phrases locate in multiple regions of a single meme, the corresponding T will include all the text information by concatenation. Let $Y = \{y_1, \ldots, y_i\}$ be the corresponding labels of all memes, where each $y \in \{0, 1\}$ with 0 means benign and 1 indicates a hateful meme. Thus, our task can be formulated as a binary classification problem with X_1 and X_2 as input. other major Objectives of this project are,

- To study and improve model performance by applying hyper parameter tuning Techniques.
- To implement better performing model using optimal parameters.

2.2 Model Design

This project is proposed a Multi Model Neural Network for this problem.Figure 2.1 shows the architecture of the proposed system.The main part of this project is feature extraction of text and image data using pre-trained models.for this the project uses two models,for text feature extraction the DistilBERT is used.DistilBERT is a transformers model, smaller and faster than BERT, which was pretrained on the same corpus in a self-supervised fashion, using the BERT base model as a teacher. This means it was pretrained on the raw texts only, with no humans labelling them in any way (which is why it can use lots of publicly available data) with an automatic process to generate inputs and labels from those texts using the BERT base model.we imported distilled BERT Model from Transformers package by HuggingFace,and the text feature is extracted from the layer before the fully connected layer.the extracted feature embedding is (,768) shaped array.

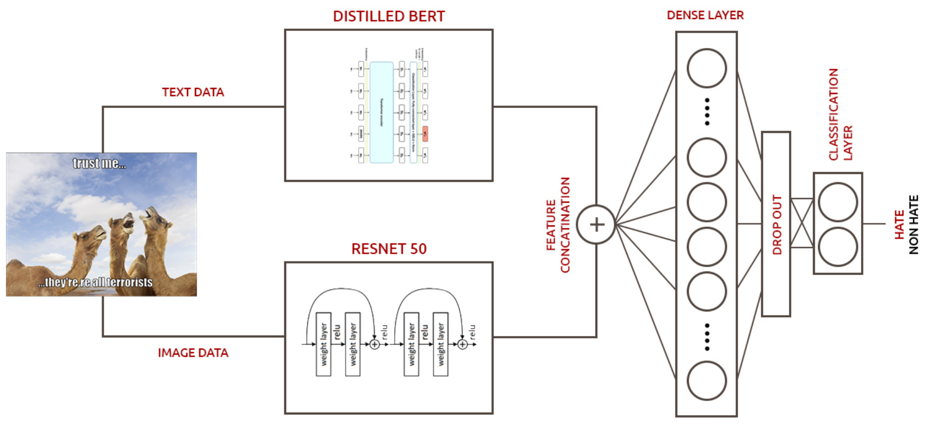


Figure 2.1: Proposed model design

For the Image feature extraction the ResNet-50 is used,ResNet-50 is a convolutional neural network that is 50 layers deep. You can load a pretrained version of the network trained on more than a million images from the ImageNet database [1]. The pretrained network can classify images into 1000 object categories, such as keyboard, mouse, pencil, and many animals. As a result, the network has learned rich feature representations for a wide range of images. The network has an image input size of 224-by-224.

The image data is fed to the pre-trained ResNet-50 model and the embeddings are collected from Cov5 layer of the ResNet model,the ResNet-50 in imported using Keras Package and model is used pre-trained weights of imagenet.After feature extraction we got (7,7,2048) sized image embedding.Before feature extraction image feature need to be flattened.so the Image features from ResNet-50 then passed to a Flatten layer,after flattening the image feature the embedding shape will be (,100352) .

After flattening of image feature embedding both image and text feature are concatenated using a concatenated layer,the output of the layer will be 100352+768 = 101120.the concatenated feature are then passed to a fully connected layer of 128/256/512 nodes with ReLu as activation function.the output of dully connected layer is then passed to a dropout layer of rate 0.2/0.3/0.5 which used to overcome overfitting.The output of dropout then will passed to fully connected classification layer which contained 2 nodes,In classification layer softmax is used as activation function.

For performance measure accuracy measure is used

2.3 Text Model

2.3.1 BERT model

BERT (Bidirectional Encoder Representations from Transformers) is paper published by researchers at Google AI Language(2020). It has caused a stir in the Machine Learning community by presenting state-of-the-art results in a wide variety of NLP tasks, including Question Answering, Natural Language Inference (MNLI), and others.

BERT's key technical innovation is applying the bidirectional training of Transformer, a popular attention model, to language modelling. This is in contrast to previous efforts which looked at a text sequence either from left to right or combined left-to-right and right-to-left training. The paper's results show that a language model which is bidirectionally trained can have a deeper sense of language context and flow than single-direction language models. In the paper, the researchers detail a novel technique named Masked LM (MLM) which allows bidirectional training in models in which it was previously impossible.

BERT makes use of Transformer, an attention mechanism that learns contextual relations between words (or sub-words) in a text. In its vanilla form, Transformer includes two separate mechanisms — an encoder that reads the text input and a decoder that produces a prediction for the task. Since BERT's goal is to generate a language model, only the encoder mechanism is necessary. The detailed workings of Transformer are described in a paper by Google. The chart below is a high-level description of the Transformer encoder. The input is a sequence of tokens, which are first embedded into vectors and then processed in the neural network. The output is a sequence of vectors of size H, in which each vector corresponds to an input token with the same index.

Masked LM (MLM)

Before feeding word sequences into BERT, 15 percentage of the words in each sequence are replaced with a [MASK] token. The model then attempts to predict the original value of the masked words, based on the context provided by the other, non-masked, words in the sequence. In technical terms, the prediction of the output words requires:

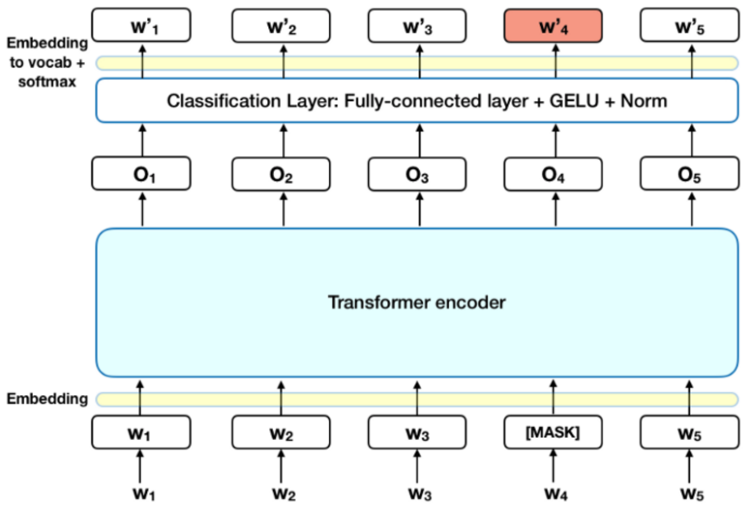

W'1
W'2
W'3
W'4
W'5
Embedding to vocab + softmax
Classification Layer: Fully-connected layer + GELU + Norm
O1
O2
O3
O4
O5
Transformer encoder
Embedding
W1
W2
W3
[MASK]
W5
W1
W2
W3
W4
W5

Figure 2.2: Masked LM

1. Adding a classification layer on top of the encoder output.
2. Multiplying the output vectors by the embedding matrix, transforming them into the vocabulary dimension.
3. Calculating the probability of each word in the vocabulary with softmax.

The BERT loss function takes into consideration only the prediction of the masked values and ignores the prediction of the non-masked words. As a consequence, the model converges slower than directional models, a characteristic which is offset by its increased context awareness.

Next Sentence Prediction (NSP)

In the BERT training process, the model receives pairs of sentences as input and learns to predict if the second sentence in the pair is the subsequent sentence in the original document. During training, 50 percentage of the inputs are a pair in which the second sentence is the subsequent sentence in the original document, while in the other

50 percentage a random sentence from the corpus is chosen as the second sentence. The assumption is that the random sentence will be disconnected from the first sentence.

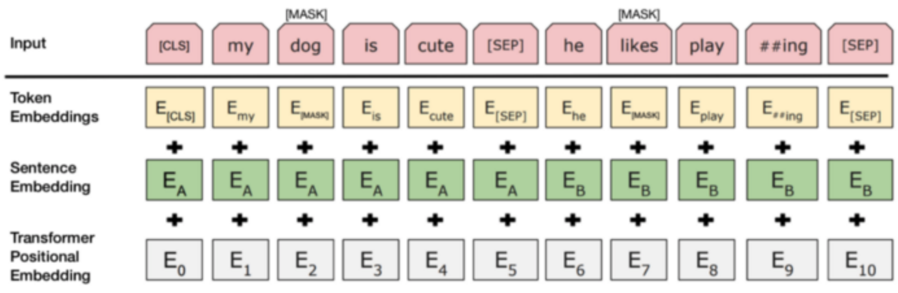

Input
[CLS]
my
[MASK]
dog
is
cute
[SEP]
he
[MASK]
likes
play
##ing
[SEP]
Token Embeddings
$E_{[CLS]}$ E_{my} $E_{[MASK]}$ E_{is} E_{cute} $E_{[SEP]}$ E_{he} $E_{[MASK]}$ E_{play} $E_{\#\#ing}$ $E_{[SEP]}$
Sentence Embedding
E_A E_A E_A E_A E_A E_A E_B E_B E_B E_B E_B
Transformer Positional Embedding
E_0 E_1 E_2 E_3 E_4 E_5 E_6 E_7 E_8 E_9 E_{10}

Figure 2.3: BERT input representation

To help the model distinguish between the two sentences in training, the input is processed in the following way before entering the model:

1. A [CLS] token is inserted at the beginning of the first sentence and a [SEP] token is inserted at the end of each sentence.

2. A sentence embedding indicating Sentence A or Sentence B is added to each token. Sentence embeddings are similar in concept to token embeddings with a vocabulary of 2.

3. A positional embedding is added to each token to indicate its position in the sequence. The concept and implementation of positional embedding are presented in the Transformer paper.

To predict if the second sentence is indeed connected to the first, the following steps are performed:

1. The entire input sequence goes through the Transformer model.

2. The output of the [CLS] token is transformed into a 2×1 shaped vector, using a simple classification layer (learned matrices of weights and biases).

3. Calculating the probability of IsNextSequence with softmax.

When training the BERT model, Masked LM and Next Sentence Prediction are trained together, with the goal of minimizing the combined loss function of the two strategies.

2.3.2 Distilled BERT

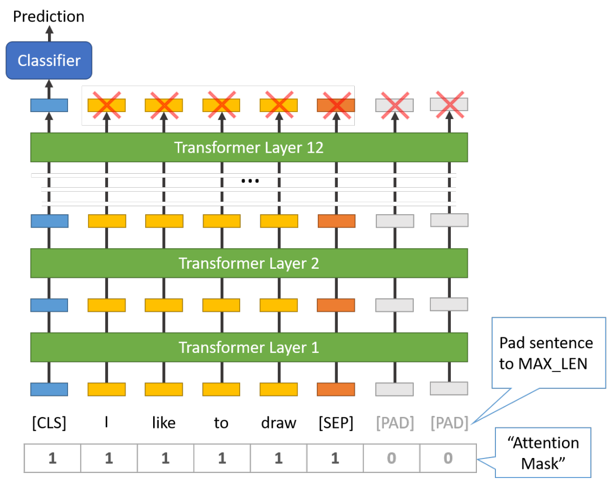


Figure 2.4: Residual Block

Distilled BERT is a smaller general purpose language representation model, which can then be fine tuned with good performances on a wide range of tasks like its larger counterparts.While most prior work investigated the use of distillation for building task-specific models, Distilled BERT leverages knowledge distillation during the pre-training phase and shows that it is possible to reduce the size of a BERT model by 40%, while retaining 97% of its language understanding capabilities and being 60% faster. To leverage the inductive biases learned by larger models during pre-training, a triple loss combining language modeling, distillation and cosine-distance losses are used.Distilled BERT is therefore a smaller, faster and lighter model and is cheaper to pre-train .

This project used Distilled BERT for Text Model feature extraction,because the maximum length of words in the text embedding from image is 80,and only small amount of data is available for training ,so it is better use Distilled BERT instead of BERT.

2.4 Image Model

2.4.1 Residual Networks

When working with deep convolutional neural networks to solve a problem related to computer vision, machine learning experts engage in stacking more layers. These additional layers help solve complex problems more efficiently as the different layers could be trained for varying tasks to get highly accurate results.

While the number of stacked layers can enrich the features of the model, a deeper network can show the issue of degradation. In other words, as the number of layers of the neural network increases, the accuracy levels may get saturated and slowly degrade after a point. As a result, the performance of the model deteriorates both on the training and testing data.

This degradation is not a result of overfitting. Instead, it may result from the initialization of the network, optimization function, or, more importantly, the problem of vanishing or exploding gradients.

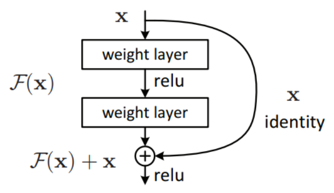

x
weight layer
$\mathcal{F}(\mathbf{x})$
relu
weight layer
x
identity
$\mathcal{F}(\mathbf{x}) + \mathbf{x}$
relu

Figure 2.5: Residual Block

ResNet was created with the aim of tackling this exact problem. Deep residual nets make use of residual blocks to improve the accuracy of the models. The concept of "skip connections," which lies at the core of the residual blocks, is the strength of this type of neural network.

These skip connections work in two ways. Firstly, they alleviate the issue of vanishing gradient by setting up an alternate shortcut for the gradient to pass through. In addition, they enable the model to learn an identity function. This ensures that the higher layers of the model do not perform any worse than the lower layers.

In short, the residual blocks make it considerably easier for the layers to learn identity functions. As a result, ResNet improves the efficiency of deep neural networks with more neural layers while minimizing the percentage of errors. In other words, the skip connections add the outputs from previous layers to the outputs of stacked layers, making it possible to train much deeper networks than previously possible.

2.4.2 ResNet-50 Architecture

- A convoultion with a kernel size of 7 * 7 and 64 different kernels all with a stride of size 2 giving us 1 layer.
- Next we see max pooling with also a stride size of 2.
- In the next convolution there is a 1 * 1,64 kernel following this a 3 * 3,64 kernel and at last a 1 * 1,256 kernel, These three layers are repeated in total 3 time so giving us 9 layers in this step.
- Next we see kernel of 1 * 1,128 after that a kernel of 3 * 3,128 and at last a kernel of 1 * 1,512 this step was repeated 4 time so giving us 12 layers in this step.
- After that there is a kernal of 1 * 1,256 and two more kernels with 3 * 3,256 and 1 * 1,1024 and this is repeated 6 time giving us a total of 18 layers.
- And then again a 1 * 1,512 kernel with two more of 3 * 3,512 and 1 * 1,2048 and this was repeated 3 times giving us a total of 9 layers.
- After that we do a average pool and end it with a fully connected layer containing 1000 nodes and at the end a softmax function so this gives us 1 layer.

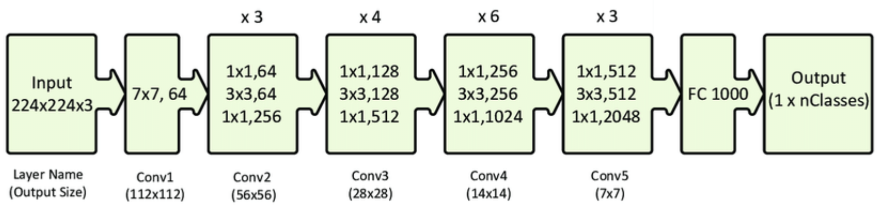


Figure 2.6: ResNet 50 Architecture

Chapter 3

Experimental Evaluation

3.1 Methodology

3.1.1 Machine Intelligence Library

Keras with Google TensorFlow backend And Huggingface's Transformers python package was used to implement the deep learning algorithms in this study, with the aid of other scientific computing libraries: matplotlib, numpy, and scikit-learn.

3.1.2 The Datasets

This project uses Facebook Hate Speech Challenge dataset which can be downloaded from Drivendata website.the 3.03 GB dataset contains 10000 images and 2 JASON files.

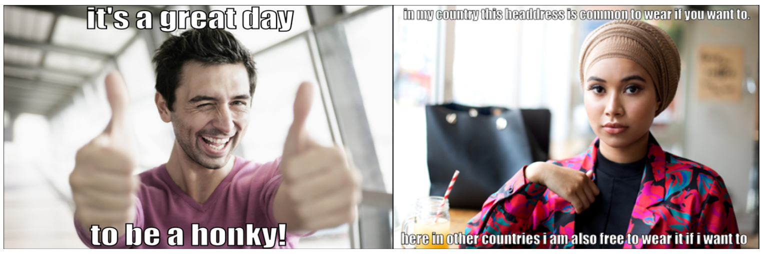

it's a great day
to be a honky!
in my country this headdress is common to wear if you want to.
here in other countries i am also free to wear it if i want to

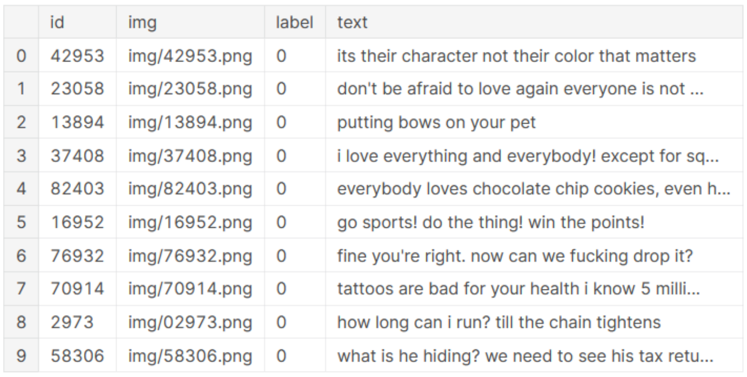
Figure 3.1: Image Data Samples

JASON files contains id,img,text and label attributes.

- **id:** Contains Integer values which is The unique identifier between the img directory and the .jsonl files, e.g., "id": 13894.
- **img:** Contains string Values which is holding actual meme file path, e.g., "img": img/13894.png, note that the filename contains the img directory described above, and that the filename stem is the id.
- **text** Contains Strings values of the raw text string embedded in the meme image, e.g., img/13894.png has "text": "putting bows on your pet"

- **label:** Contains Integer values,0 and 1 ,the 1 indicates meme is "hateful" and 0 indicates "non-hateful"

The Training and Validation split is 90 and 10 respectively, and the individual sets are fully balanced. Each meme in the training and validation set are annotated as either 1 or 0 which corresponds to the hateful and non hateful classes respectively.



	id	img	label	text
0	42953	img/42953.png	0	its their character not their color that matters
1	23058	img/23058.png	0	don't be afraid to love again everyone is not ...
2	13894	img/13894.png	0	putting bows on your pet
3	37408	img/37408.png	0	i love everything and everybody! except for sq...
4	82403	img/82403.png	0	everybody loves chocolate chip cookies, even h...
5	16952	img/16952.png	0	go sports! do the thing! win the points!
6	76932	img/76932.png	0	fine you're right. now can we fucking drop it?
7	70914	img/70914.png	0	tattoos are bad for your health i know 5 milli...
8	2973	img/02973.png	0	how long can i run? till the chain tightens
9	58306	img/58306.png	0	what is he hiding? we need to see his tax retu...

Figure 3.2: JASON file Data samples

3.1.3 Data Preprocessing

While Exploring the text embeddings from image meme,

- Found presence of characters
- Found presence of punctuation
- Found presence of Extra spaces
- Found presence of Digits

So we removed characters,extra spaces and digits using python RE package and also we converted punctuation to its corresponding words.for the Image data we re-sized image data according to ResNet-50 Architecture.for ResNet-50 we have to input 224*224*3 size image data.

3.1.4 Hyper Parameter Tuning

In machine learning, hyper parameter optimization or tuning is the problem of choosing a set of optimal hyper parameters for a learning algorithm. A hyper parameter is a parameter whose value is used to control the learning process. By contrast, the values of other parameters (typically node weights) are learned. For this project's Hyper parameter tuning following parameters are considered,

- **Dropout:** Dropout is a regularization technique for neural network models proposed by Srivastava, et al. in their 2014 paper Dropout: A Simple Way to Prevent Neural Networks from Overfitting (download the PDF).Dropout is a technique where randomly selected neurons are ignored during training. They are "dropped-out" randomly. This means that their contribution to the activation of downstream neurons is temporally removed on the forward pass and any weight updates are not applied to the neuron on the backward pass. For this project 0.2,0.4 and 0.5 Dropout rates are used for Hyper parameter Tuning.

- **Batch Size:** The batch size is a hyper parameter that defines the number of samples to work through before updating the internal model parameters.Think of a batch as a for-loop iterating over one or more samples and making predictions. At the end of the batch, the predictions are compared to the expected output variables and an error is calculated. From this error, the update algorithm is used to improve the model, e.g. move down along the error gradient. For this project 16,32 and 64 batch sizes are used for Hyper parameter Tuning.

- **Number of Nodes in Hidden Layer:** In this project a fully connected layer used after concatenation Layer for image and text feature concatenation.fully connected layer used different number of nodes in hyper parameter tuning For this project 128,256 and 512 Number of nodes are used in hidden layer for Hyper parameter Tuning.

3.1.5 Optimization and Loss

Adam optimization is an extension to Stochastic gradient decent and can be used in place of classical stochastic gradient descent to update network weights more efficiently.For this project ADAM optimizer is used as optimization algorithm.

In this project Binary Cross Entropy function used as loss function for calculating error of the model.Binary cross entropy compares each of the predicted probabilities to actual class output which can be either 0 or 1. It then calculates the score that penalizes the probabilities based on the distance from the expected value. That means how close or far from the actual value.

$$\text{Log loss } = \frac{1}{N} \sum_{i=1}^{N} -\left(y_i * \log\left(p_i\right) + \left(1 - y_i\right) * \log\left(1 - p_i\right)\right) \tag{3.1}$$

Here, pi is the probability of class 1, and (1-pi) is the probability of class 0.

3.2 Results

The results are shown in Table 3.1. We observed that the The Distilled BERT using Text model performed better than ResNet-50 used Image model while comparing the uni models.but while comparing uni model with mutlimodel,the concatenation model which used both Distiled and ResNet-50,The multi model performed better than uni models.

MODEL ACCURACY	**ALGORITHM**	**VAL ACCURACY (Percentage)**
Text Only Model	DistilBERT	60
Image Only Model	ResNet 50	57
Text + Image Model	Concatenation Model (DistilBert+Resnet50)	67

Table 3.1: Model Comparison

The hyper parameter tuning results shown in Table 3.2.We observed that model 1 performed better than other models.In model 1 the used hyper parameters are Hidden Layer Units = 256, Batch Size = 16 And Drop Out = 0.4.

Model No	Hidden Layer Units	Batch Size	Drop Out Rate	Validation Accuracy
1	256	16	0.4	67
2	256	32	0.5	65
3	128	16	0.2	65
4	512	16	0.5	63

Table 3.2: Hyper Parameter Tuning Results

3.3 Discussion

It can seen that model trained on the facebook challenge dataset has an accuracy 67% on prediction of validation data which is better than the text only model and image only model which trained on Distilled BERT and ResNet-50 Respectively.from this observation we can say the the multi model is the best performing model than the uni models.

Chapter 4

Related Work

Hate speech detection has gained more and more attentions in recent years. There have been several text-only hate speech datasets released, mostly based on Twitter [(Waseem, 2016),(Waseem & Hovy, 2016),(Davidson et al., 2017)], and various architectures have been proposed for classifiers [(Kumar et al., 2018), (Malmasi & Zampieri, 2017)]. Also, in the past few years, there has been a surge in multi-modal tasks and problems, ranging from visual question answering[(Goyal et al., 2017)] to image captioning[(Sidorov et al., 2020),(Gurari et al., 2020)] and beyond. However, there has been surprisingly little work related to multi-modal hate speech, with only a few papers including both image and text modality.

Some of the works related to multi-modal hate detection based on image and text modality are as follows.Yang et al. [(Yang et al., 2019)] reported that augmenting text with image embedding information immediately foosts performance in hate speech detection. In this paper, he image embeddings are formed by using the second last ayer of the pre-trained ResNet neural network on ImageNet mage and text vectors. The concatenated vector is followed by dropout, MLP, and softmax operations for the final hate peech classification. They also explore other fusion techniques like gated summation and bi-linear transformation.

Gomez et al.(Gomez et al., 2020) highlighted the issue that nost of the previous work on hate speech is done using texpublications has not been addressed yet. So, they created taining text and an image. The data points are labeled into one of the six categories: No attacks to any community, by proposing two models. The first one was the Featun Concatenation Model (FCM). which is catenates the image representation extracted by a CNN anc he textual features of both the tweet text and the image he image corresponding to the associated texts. This was lone by learning kernels from textual representations and involving them with CNN feature maps.

Chapter 5

Conclusion and Future Works

This projected implemented multi model deep learning system for detecting hate speech in memes using DistilBERT and ResNet-50 Network embeddings.our project has fine-tuned the model with an extra layer of neurons to specifically serve the purpose of classifying review text into fake or genuine class. Creating a completely new model from scratch to do the same purpose would never have resulted in such a good accuracy. Thus we leveraged the pre-trained model to the best of its ability.

In This project we done hyper parameter tuning using number of nodes in hidden layer,batch size and drop out layer rate,and got best performing parameters.and implemented model using best performing parameters. Compared to the existing multi models our model shows lesser accuracy.existing models like VISUAL-BERT and VILBERT shows better accuracy than our model.

Our model can improve accuracy by applying more data into dataset,can increase data using data augmentation in future.and also improve model accuracy by adding model features from image data into text dataset like captions from image or objects from images.

Bibliography

[1] Douwe Kiela, *The Hateful Memes Challenge:Detecting Hate Speech in Multimodal Memes*, NeurIPS, 2020.

[2] Jacob Devlin, *BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding*, Google AI Language, 2018 .

[3] Victor Sanh, *DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter*, NeurIPS, 2019.

[4] Kaiming He, *Deep Residual Learning for Image Recognition*, Microsoft, 2015 .

[5] Chi Sun, *How to Fine-Tune BERT for Text Classification?*, 2020 .

[6] A. Sai Bharadwaj Reddy, *Transfer Learning with ResNet-50 for Malaria Cell-Image Classification*, 2020 .

Appendix - A

Abbreviations

BERT	-	Bidirectional Encoder Representations from Transformers
CNN	-	Convolutional Neural Network