Birla Institute of Technology and Science Pilani

Neural Networks And Fuzzy Logic

Report

Prepared by

Shah Het Divyangkumar	2017A7PS0093G
Avishree Khare	2017A7PS0112G
Ashwin Vaswani	2017A7PS0960G

Pre Processing :

For text, we have removed URLs specifically those words ending with ".com" and ".net" and also the ones starting with "www." as most of the text contained the links from which the memes were obtained which was not needed for classification. We also performed some other basic text cleaning steps such as stopword removal, lower casing etc. We have further used Glove Embedding for the text input to our deep neural net. For images, we have just reshaped them to (299,299,3) without any further preprocessing. We have dropped five rows from the dataframe as they contained corrupt images. We have changed some values in the predicted columns as they were misplaced.

Methodology :

Our first take on the project was using **Multi-task learning (MTL)**. Although this approach was quite misleading in the sense that it was not expected of us, we did learn a lot about generalizing deep learning using Multi-task models. We tried both the hard parameter sharing and the soft parameter sharing techniques of MTL.

The next approach we took was training models specific to either text or image data. We used **ImageNet pre-trained** models for classifying images. Text data were analyzed using standard sequence models, i.e., **word embeddings, LSTMs and GRU**. Some Machine Learning models including **XGBoost** were also considered. We also trained a few advanced transformers including **BERT** and **XLNet** on text data. We, however, realized that such unimodal architectures were incapable of capturing the fine details that were present in the other modalities.

Our pre-final approach is attributed to the double **additive paradigm** which incorporates the properties of additive combination. We believed that this model (Figure. 1) encapsulated the features from both modalities (text and images, here) by bridging the semantic gap using late fusion while preserving individual properties from branching independent layers. The model, however, did not perform well on the dataset, reasons for which, we shall present in later sections.

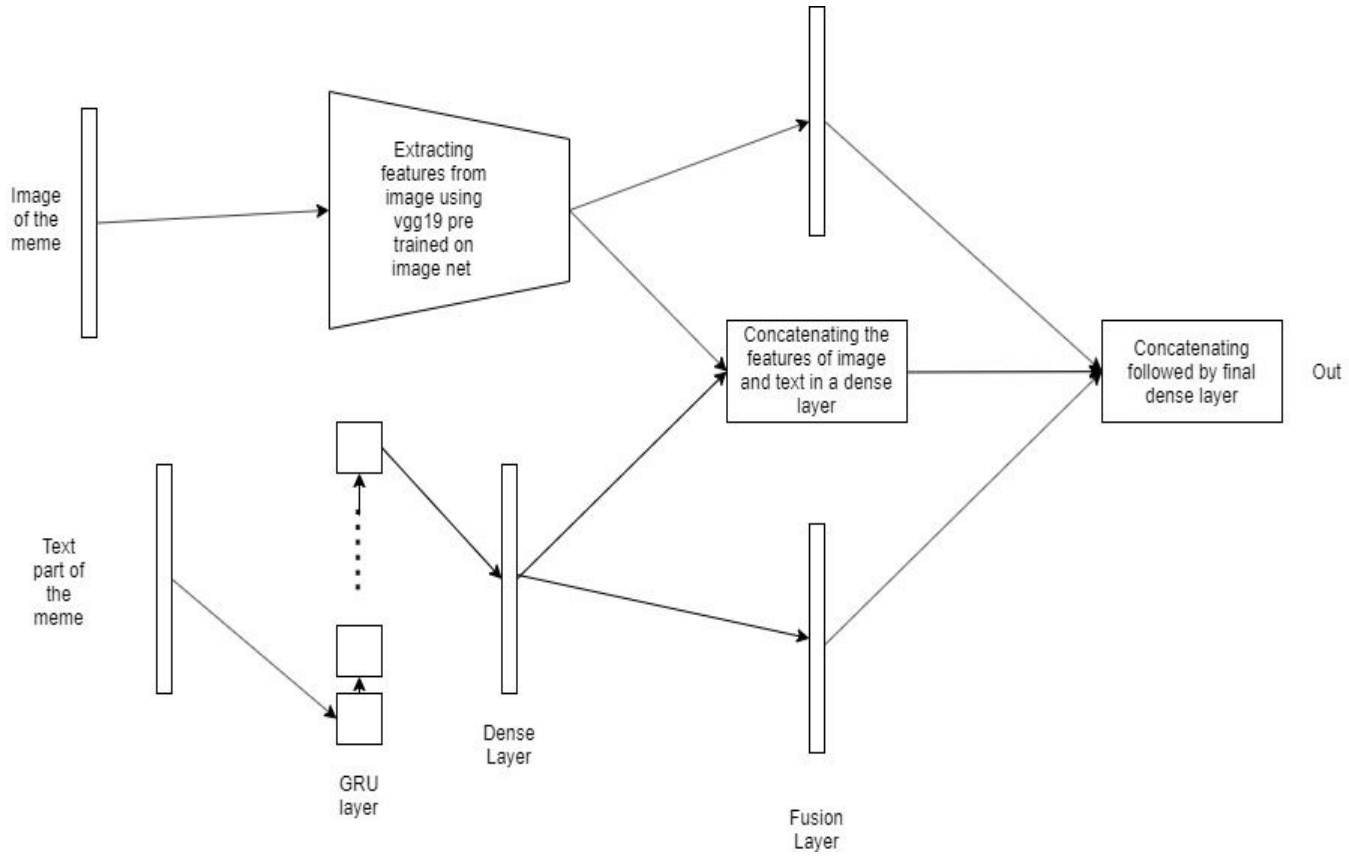


Figure. 1. Double additive model

Our final model is based on the idea of **Additive combination**. Being one of the traditional approaches for multimodal classification, this model was considered as it boosted the F1 score. We shall present the details of this model in the next section.

The following document lists the notable approaches we encountered and their results. We have also tried to explain the results and the shortcomings of our work.

Approaches :

1. **XGBoost :**

   The first approach was a basic machine learning model to classify text. We used XGBoost for this purpose as it is known to give good results on textual classification. We did some basic text preprocessing, stemming and dropped rows with NA values in the sentiment column. We then used BagOfWords to create word vectors which were used as input to the classifier. We trained the model with default parameters. The confusion matrix is shown below :

   [[0, 407, 0, 0, 0],
   [0, 148, 0, 0, 0],
   [0, 272, 0, 0, 0],
   [0, 50, 0, 0, 0],
   [0, 20, 0, 0, 0]]

   The F1 score was 0.081793 with precision being 0.160196 and recall being 0.127256. The accuracy of this model was 45.0766%. The model did not perform well at all and this is where we had a suspicion about something being wrong.

2. **GRU model for text classification :**

   We further tried benchmarking the dataset using a simple GRU model along with glove embedding. We used the same preprocessing that we did in the last part. We used the following model:

   The confusion matrix was :

   [[0, 102, 0, 0, 0],
   [0, 304, 0, 0, 0],
   [0, 206, 0, 0, 0],
   [0, 35, 0, 0, 0],
   [0, 13, 0, 0, 0]]

   The F1 score was 0.126141 with precision and recall being 0.092121 and 0.2

respectively and the accuracy was 44.021%.

3. **Double Additive model :**

   We built a model incorporating both the visual and textual data from the dataset. The model used a pre-trained InceptionResnetV2 for feature extraction from images while the text was processed using GloVe embeddings and GRU.

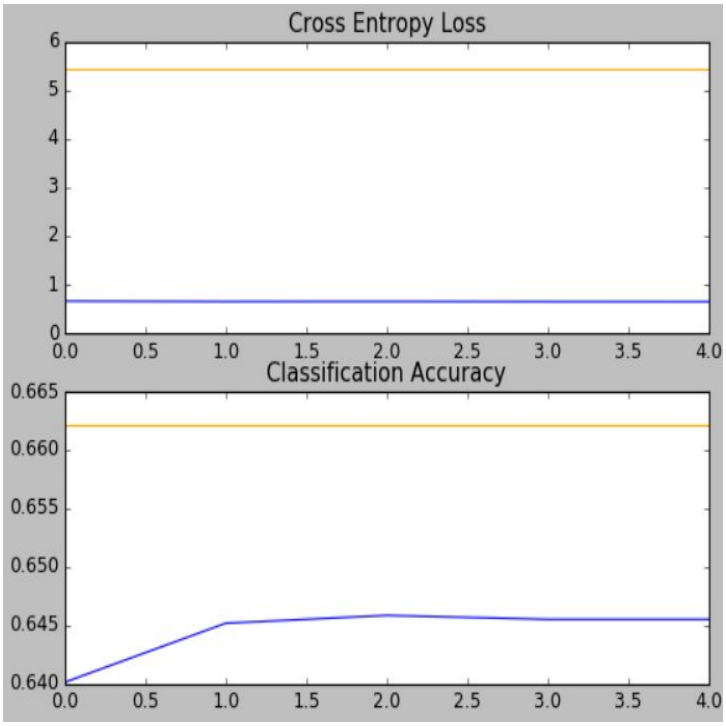
   This model employs the concept of additive combination for bridging the semantics between the two modalities while extracting distinct features from them independently.

   We tried different configurations of hyperparameters for the three tasks. In general, we maintained a batch size of 64 and trained on 15 epochs. We experimented with setting class weights too as the targets were biased.

TASK A: Motivational Vs. Not motivational :

The model achieved an F1 score of 0.398359, precision of 0.331061, a recall of 0.50 and an accuracy of 0.662121 on the validation set. At the end of five epochs, the validation loss was 5.44.

The loss and accuracy curves are as follows:



The confusion matrix obtained is as follows:

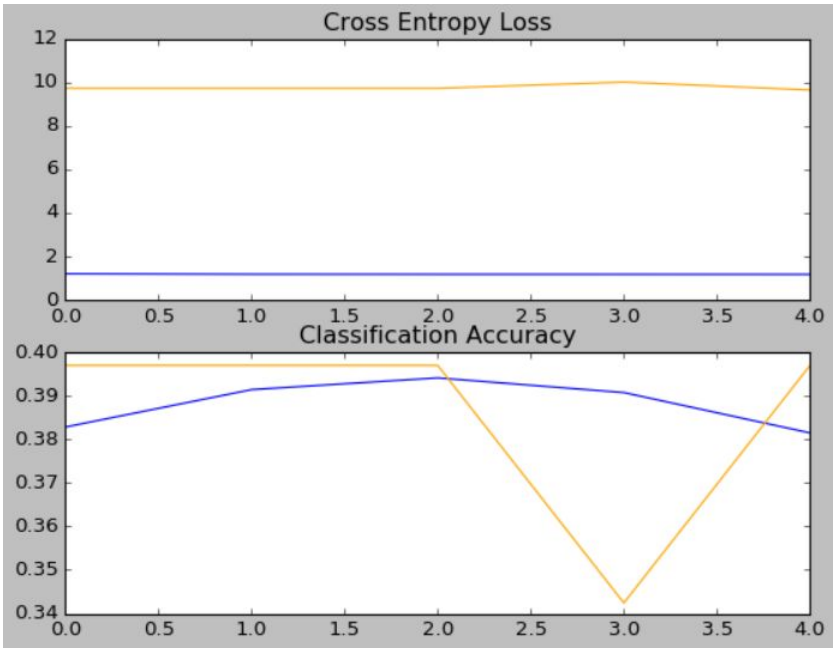
[[440, 0]

[220, 0]]

TASK B: Offensiveness classification :

The model achieved an F1 score of 0.142802, a precision of 0.099242, a recall of 0.25 and an accuracy of 0.396970 on the validation set. At the end of five epochs, the validation loss was 9.6432.

The curves corresponding to loss and accuracy are as follows:



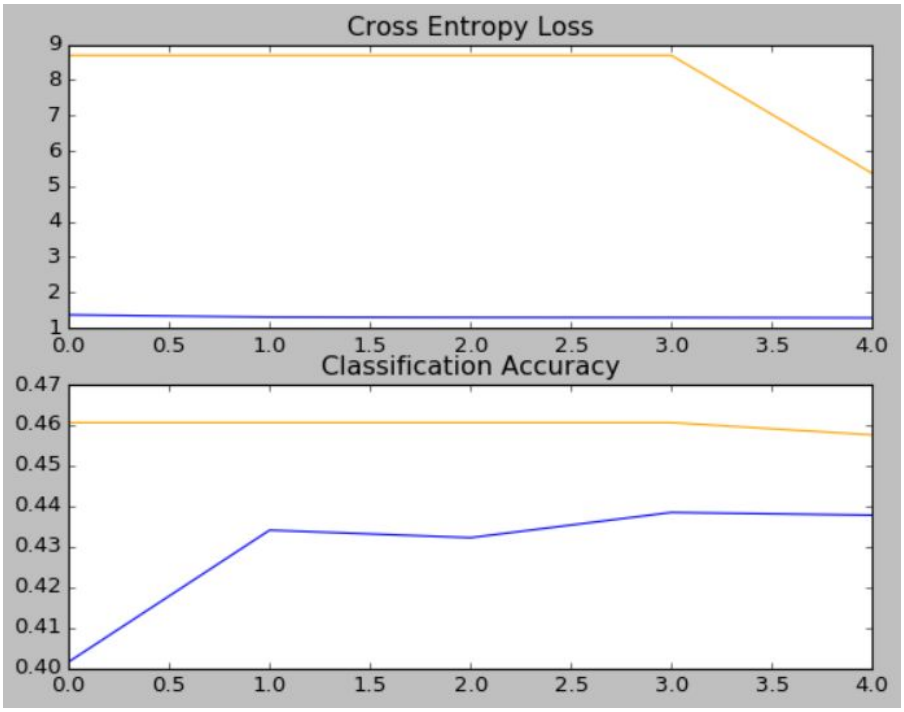
The confusion matrix obtained is as follows:

[[262, 0, 0, 0],
[226, 0, 0, 0],
[151, 0, 0, 0],
[21, 0, 0, 0]]

TASK C: Sentiment Classification :

The model achieved an F1 score of 0.164162, precision of 0.174390, recall of 0.207761 and accuracy of 0.457576 on the validation set. At the end of five epochs, the validation loss was 5.5444.

The loss and accuracy curves were as follows:



The confusion matrix created was as follows:

[[0, 95, 7, 0, 0],
[0, 273, 31, 0, 0],
[0, 177, 29, 0, 0],
[0, 32, 3, 0, 0],
[0, 12, 1, 0, 0]]

4. **Additive Combination :**

Our current model uses the concept of additive combination. We have used VGG16 pre-trained on ImageNet for image feature extraction and GloVe embeddings for processing text. We then have a concatenation layer to combine the features before feeding the final classifier. Although this setup

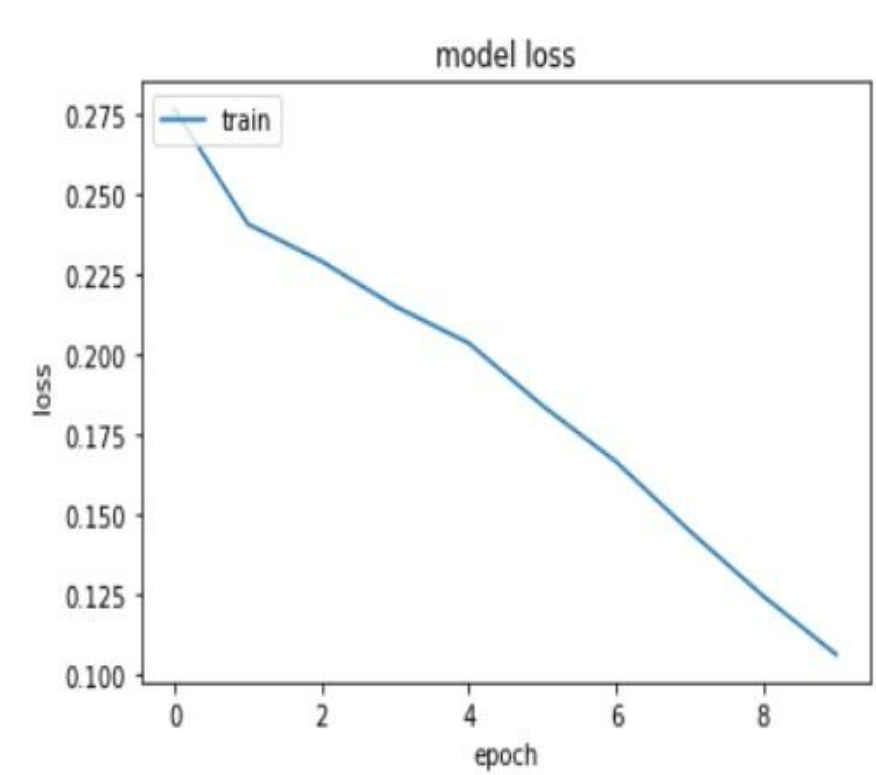
remains similar for all three tasks, there are subtle inclusions and deletions. We have used Adam optimizer with default parameter values and have trained for an average of 15 epochs.

TASK A: Motivational Vs. Not motivational :

The model achieved an F1 score of 0.810456, precision of 0.806244, a recall of 0.816126 and an accuracy of 0.824242 on the validation set.

```
F1.........: 0.810456
Precision..: 0.806244
Recall.....: 0.816126
Accuracy...: 0.824242
```

The training loss curve is as follows:



The confusion matrix created is as follows:

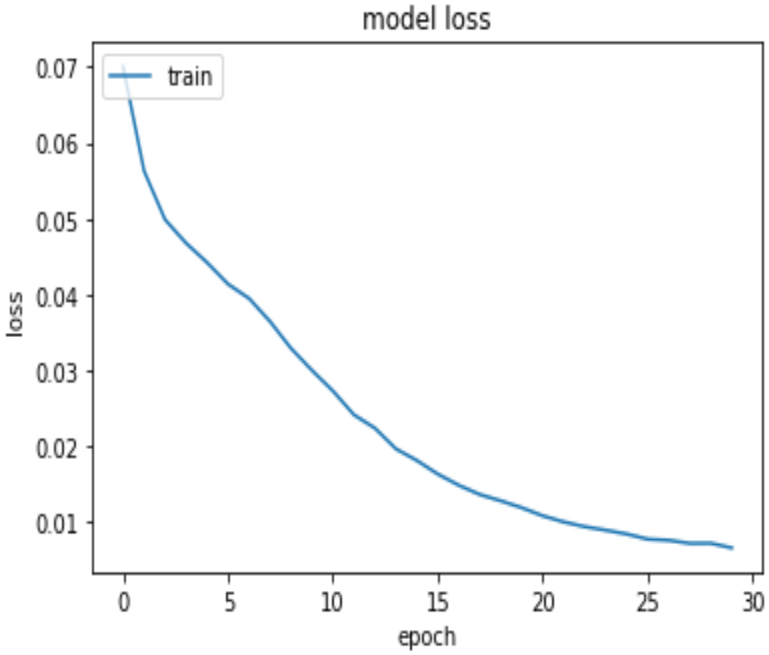
```
array([[361,  67],
       [ 49, 183]])
```

TASK B: Offensiveness Classification :

The model achieved an F1 score of 0.2353273, precision of 0.269627, a recall of 0.258194 and an accuracy of 0.384848 on the validation set.

```
F1.........: 0.235273
Precision..: 0.269627
Recall.....: 0.258194
Accuracy...: 0.384848
```

The training loss curve is as follows:



The confusion matrix created is as follows:

```
array([[170,  76,  11,   1],
       [163,  75,   5,   0],
       [ 92,  37,   9,   0],
       [ 11,   5,   5,   0]])
```

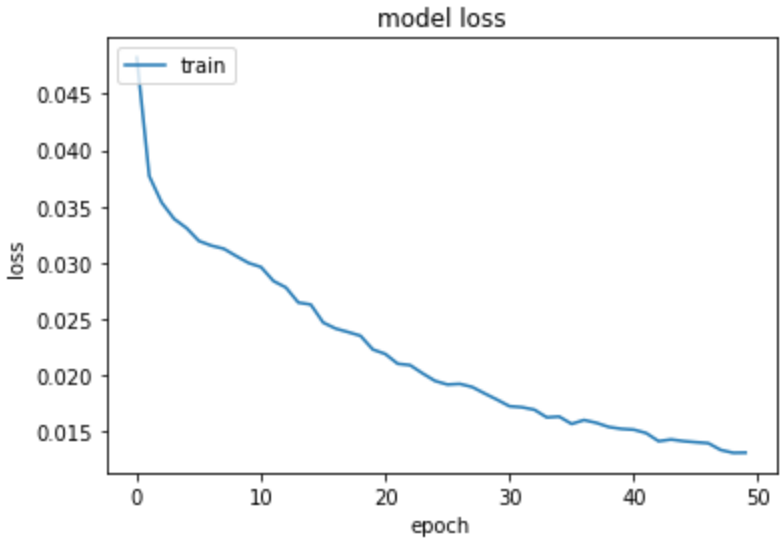
The MAE was found to be 0.81.

TASK C: Sentiment Classification :

The model achieved an F1 score of 0.222885, a precision of 0.222262, a recall of 0.226211 and an accuracy of 0.375757 on the validation set.

```
F1.........: 0.222885
Precision..: 0.222262
Recall.....: 0.226211
Accuracy...: 0.375758
```

The loss curve is as below:



The confusion matrix created is as follows:

```
array([[ 21,  45,  29,   2,   0],
       [ 42, 148,  94,  11,   0],
       [ 32,  91,  77,   9,   0],
       [  9,  18,  15,   2,   1],
       [  0,  11,   3,   0,   0]])
```

Problems faced :

1. The dataset given to us had a considerably high variance. To overcome that we trained the dataset on a simpler network. Considering the bias-variance trade-off, we used dropout to regularize the model and also initialized the weights of the dense layers using a random normal distribution.
2. We trained the dataset on VGG, ResNet and Inception-ResNet. Inception-ResNet performed well on a similar task[1] but didn't work well on this dataset. VGG19 gave us the best results among all the pre-trained models.
3. The model predicted only one class. To overcome this, we regressed instead of categorical classification and then the model started to predict other classes. Although, as we can see in the task of offensiveness, it still didn't predict many of the other classes.

References :

[1] https://arxiv.org/abs/1906.08595

[2] https://www.aclweb.org/anthology/D18-1382.pdf

[3] https://arxiv.org/abs/1805.11730

[4] https://arxiv.org/ftp/arxiv/papers/1012/1012.5994.pdf

[5] https://ieeexplore.ieee.org/document/8354676

[6] https://arxiv.org/pdf/1801.04433.pdf

[7] https://arxiv.org/pdf/1805.10205.pdf