Introduction

In this work, we investigate the effectiveness of importance weighting for highly imbalanced data. We analyze this for a variety of deep learning architectures, tasks and datasets. For the experiments we are mainly focusing on classification tasks. We analyze the impact of importance weighted loss function for scenarios when there is Label Shift in the imbalance data. For this our experiments address both the classification of image dataset using Flower Recognition Data, and Text dataset using Recognizing Textual Entailment (RTE) (Wang et al. #) data, and The Stanford Sentiment Treebank (SST-2) data (Socher et al. #). The authors in the paper (Byrd and Lipton #) have studied the effect of importance weighting on neural networks by down-weighting the loss contributions of examples from a particular class. They observed that as the training progresses, the effects of importance weighting vanishes. We wanted to study if this performance persists over all the different datasets and scenarios. In addition to that, we wanted to know if the IPW effect stays for classification tasks that do not traditionally have high accuracy (e.g. RTE dataset).
We extend this idea for comprehending the effect of normalized importance weighted loss function for Online Deep Learning Training Scenario using 2-D synthetic dataset.

Experiment

We performed several experiments to investigate the effects of importance weighted loss function on neural networks for Flower image dataset, RTE (Wang et al. #) and SST-2 text dataset (Socher et al. #). Experiments using these datasets address the label shift scenario, where we are weighting the loss function based on the class fractions of the training samples. For eg. If class 1 has a proportion of 0.1 in the dataset whereas class 2 has a proportion of 0.9, we weight the loss function of class 1 with a fraction of 0.9 and class 2 with a fraction of 0.10. We hypothesize that using the importance weighted loss function would lead to faster convergence of the Deep learning model and would provide us an improved accuracy score than the unweighted Cross Entropy loss.

In addition to that, we performed experiments using 2-D synthetic dataset to analyze the effect of normalized importance weighted loss function.

Synthetic 2-D Dataset

To understand the effect of ipw normalization on convergence of Machine Learning models, we trained a Logistic Regression model using Stochastic Gradient descent optimization on a 2D imbalanced synthetic data. For a binary classification task, the Cross Entropy loss can be given by:

$$BCE\ loss = \frac{1}{Batch\ size} \sum_{i=1}^{Batch\ size} -\left\{ \frac{y_i log(\mathcal{F}(x_i; w))}{\pi} + \frac{(1 - y_i) log(1 - \mathcal{F}(x_i; w))}{(1 - \pi)} \right\}$$

Taking a leaf out of ipw normalization, we formulated the loss function as:

$$BCE_{normalised\ ipw} = \frac{\sum_{i=1}^{Batch\ size} -\left\{ \frac{y_i log(\mathcal{F}(x_i; w))}{\pi} + \frac{(1-y_i) log(1-\mathcal{F}(x_i; w))}{(1-\pi)} \right\}}{\frac{1}{\pi} * \Sigma \mathbb{1}(y_i = 1) + \frac{1}{(1-\pi)} * \Sigma \mathbb{1}(y_i = 0)}$$

We hypothesize that the normalized loss function should work better in case of highly imbalanced data and in an online training setting. The problem with using just the weighted Cross entropy loss is the high variability in the loss values. For eg. In a scenario when in a batch there are many minority class samples, using the cross entropy loss would give a huge loss value whereas if there are many majority class samples, the cross entropy loss would result in a low loss value. Whereas, normalizing the loss by weighting the number of samples from each class would give us a more stable loss function and might lead to faster convergence.

Adaptive optimization methods such as Adam optimizer have been proved to work better and helped for faster convergence. We wanted to observe the effect of Adam optimizer vs the effect of normalized loss and also comprehend the effect of variance in the loss values.

Results:

We observe a high variability in the loss values for Cross Entropy loss whereas there is very little variance in ipw normalized loss. Also, the loss values for Cross Entropy loss

are quite high for every epoch, whereas for the normalized ipw loss, the loss values after the initial epoch decreases much faster.

Also, the experiment was modeled as an online training setting so each batch could have any number of positive and negative samples. Hence, we used metrics like F1 score to compare the performance of the two models. The average F1 score for weighted loss and Adam optimizer for different random seeds and different epochs is 0.62 whereas for the unweighted loss it is approximately 0.5. The final F-1 score for weighted loss and Adam optimizer is approximately the same whereas for the unweighted score, it is very less.

We observe that the normalized weighted loss works quite well in comparison to the unweighted loss. After a certain number of epochs, the convergence is way better for normalized weighted loss in comparison to the unweighted loss.

Conclusion:

1. The normalized ipw loss provides stable training as compared to the unweighted loss and the Adam optimizer.
2. When the F-1 score or accuracy is compared the Adam optimizer performance is on par with the normalized ipw loss. However, the unweighted loss does not have as significant performance as the other two.
3. There is a high variability in the unweighted loss for each epoch and every random seed, which might lead to taking huge steps and time to converge. Also, the training loss does not decrease gradually and is variable.
4. Thus, we observe a potential for using normalization for online training tasks that have highly imbalanced data.

Label Shift Experiments

Flower Image Dataset

Data Description

The flower dataset contains 4242 images of flowers. There are 5 classes in the dataset: chamomile, tulip, rose, sunflower, dandelion. Each class has approximately 800 images. Each image is of the size 320 x 240 pixels.

We conduct mainly two experiments with the dataset:

Experimentation 1:

Given the 5 flower classes, we provide the classes in different ratios to understand the effect of importance weighted loss function on a highly imbalanced dataset. We performed the experiment using the VGG16 pre-trained and non-pretrained model. We take a proportion of samples [1/16, 1/8, 1/5, 1/2, 1] from the 5 flower classes. The Cross Entropy loss is thus weighted in the order [16, 8, 4, 2, 1]. The test samples contain the flowers of each class equally. We compare these results to the unweighted Cross Entropy loss.

Results:

We have a much improved accuracy score for the importance weighted loss in comparison to the unweighted loss. It is evident from the loss and train accuracy of each epoch that the importance weighted model converges faster than the unweighted model. We can clearly observe that penalizing the minority class with a higher value than the majority class has helped us improve the accuracy of our model.

Experimentation 2:

Given 5 flower classes and a binary classification task where rose and tulip belong to one class and chamomile, sunflower and dandelion in the other class, and if there is an imbalance within one class, (ie. 80:20 ratio of rose:tulip), we wanted to observe the importance of ipw loss. Whenever a tulip example was encountered and misclassified, we highly penalize that instance (ie. up-weight the loss for tulip labels), whilst keeping the loss for other samples to be the inverse of their proportion of occurrence in the training dataset. For the validation set, we kept the number of samples of each flower to be the exact same. Hence, it was possible to use accuracy as the metric to compare the unweighted and weighted loss.

Results:

Faster convergence: The weighted loss leads to faster convergence when the accuracy and loss for every epoch is compared.
Accuracy improvement: There is a huge difference in the overall accuracy value for the weighted loss (37% for the 5 class) as compared to the unweighted loss (21%).

Conclusion:

1. Weighted loss performs better for both the classification experiments. Moreover, the convergence is faster as compared to the unweighted loss function.
2. The importance weighted loss can be a good substitute for data augmentation in cases when there are limited computational resources.
3. On running the experiments for different imbalanced datasets, we observed that more the imbalanced in data, better the effect of the ipw estimator.

Text Dataset

For the experiments using the text dataset we are using RTE (Wang et al. #) dataset and SST-2 dataset. Sections below describe the datasets, models used and results for both the datasets.

Data Description

1. **RTE-** RTE (Wang et al. #) is an inference task data that come from a series of annual textual entailment challenges and examples are based on news and Wikipedia text. It is a two-class dataset, with two class labels- **entailment: 0**, and **non_entailment: 1** where neutral and contradiction classes are collapsed into not entailment.
2. **SST-2-** SST-2 (Socher et al. #) is a classification task data consisting of sentences from movie reviews and human annotations of their sentiment. It is a 2-class dataset , with two class labels- **negative: 0**, and **positive: 1** and the task is to predict the sentiment of a given sentence. Here we only use the sentence level sentiment.

Experimentation

For the classification of text data into any one of the classes we are using the BERT model (Devlin et al. #). We are running the experiments when the loss function is weighted according to the equation and non-weighted loss function for 5 different fractions of class 1 data- 0.1, 0.25, 0.5, 0.75, and 1, i.e. we run these experiments by varying the fraction of class 1 data while all the data samples from the class 0 are taken during the training. These experiments are performed with both pretrained and non-pretrained BERT models. We are running these experiments for 15 epochs with a learning rate of 0.0001.

Results

To analyze the effectiveness of importance weighted loss function in convergence and model performance we compared the validation accuracy and validation loss of importance weighted loss function vs non-importance weighted loss function for all the five different fractions of class 1 data. We compared these results separately for both pre trained and non-pretrained BERT models.

RTE

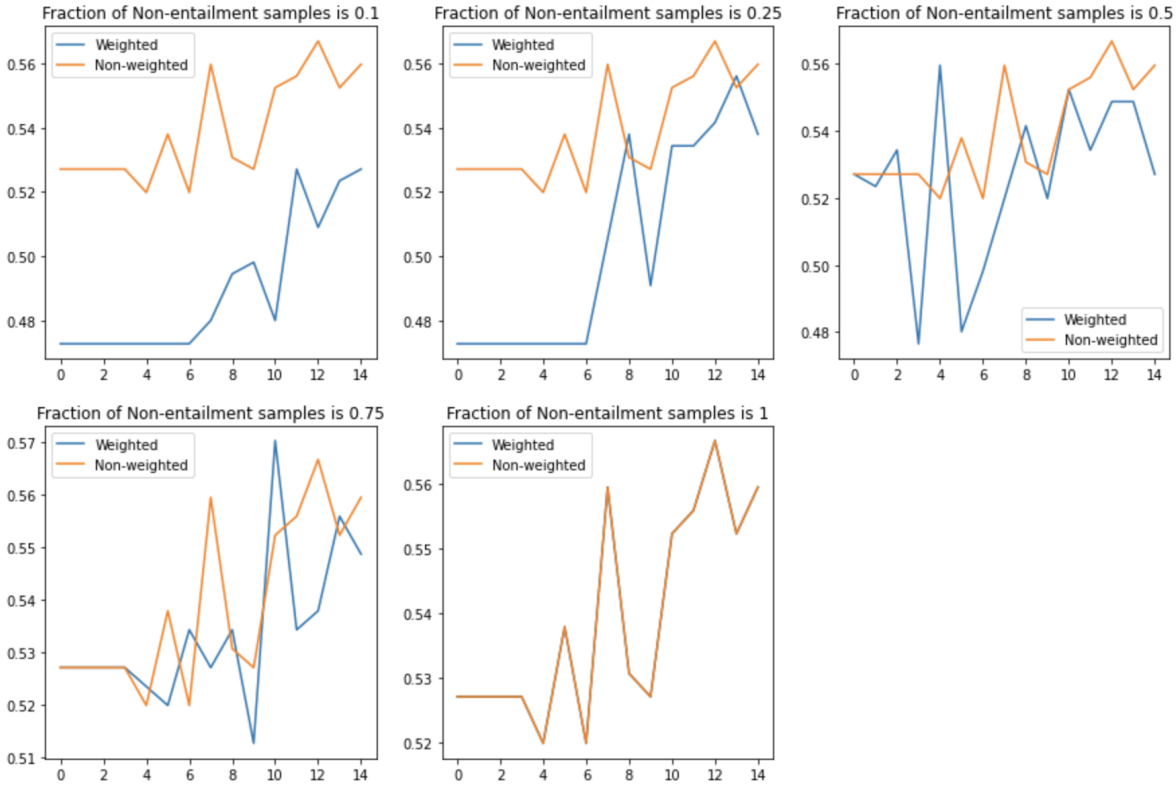
In this section we compare the performance in terms of validation accuracy and loss for the RTE dataset.

1. Non-Pretrained BERT model

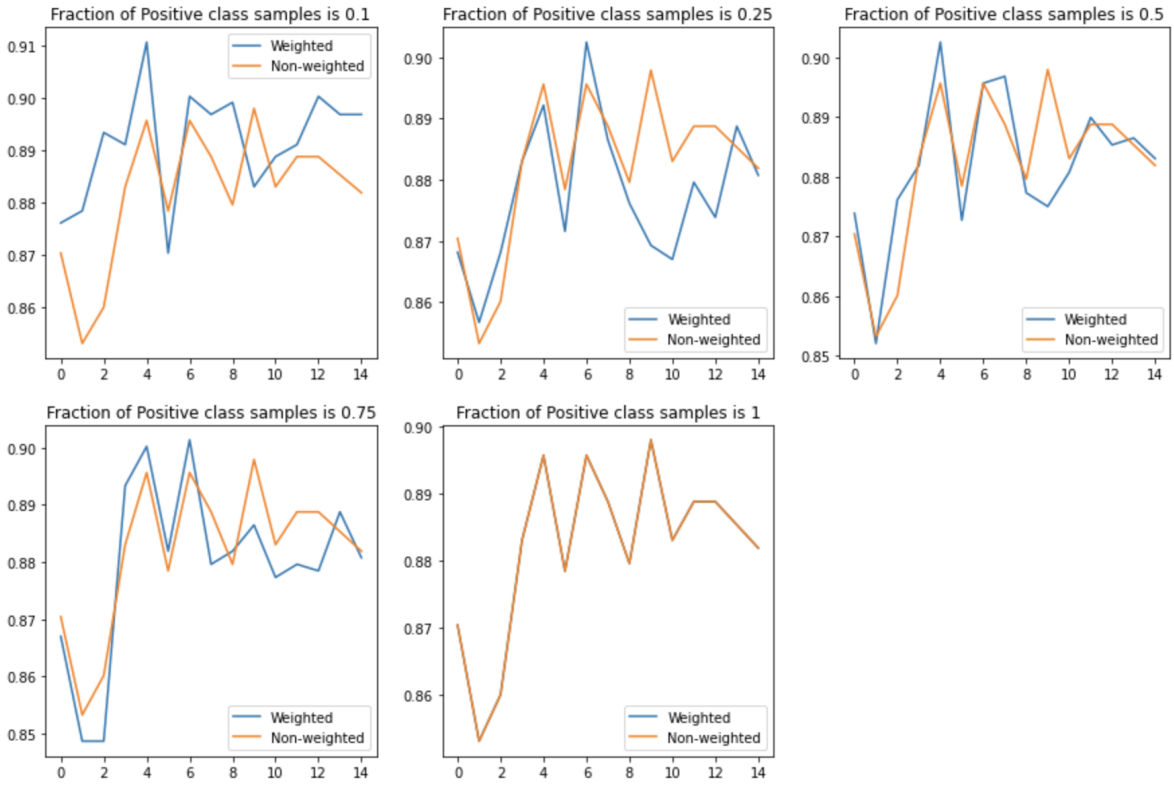
For non-pretrained BERT models the validation accuracy of non-weighted loss function is higher when there is high data imbalance (i.e. the fraction of class 1 or non_entailement is 0.1 or 0.25) for all the epochs. This difference in the validation accuracy score reduces as the fraction of class 1 or non_entailement data increases in the training. This observation deviates from our hypothesis “weighted loss function gives better performance than the unweighted loss function with higher accuracy”. Also we can see that the validation loss for the weighted loss function is higher than the unweighted loss function in initial epochs. This observation also deviates from our hypothesis “weighted loss function gives faster convergence than the unweighted loss function”.

This might be because given the model complexity it does not have enough data to learn anything meaningful and does not give better performance. Also from the accuracy score we can see that the model is not learning anything meaningful as the score remains near 0.5 even when trained using a full dataset and it seems like it is giving predictions randomly.

a. Validation Accuracy



b. Validation Loss

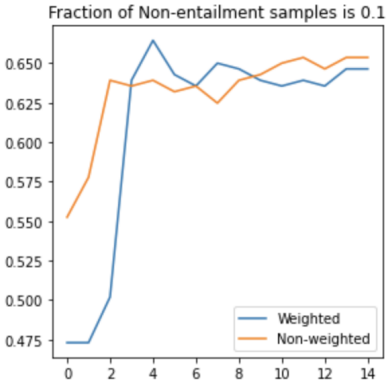


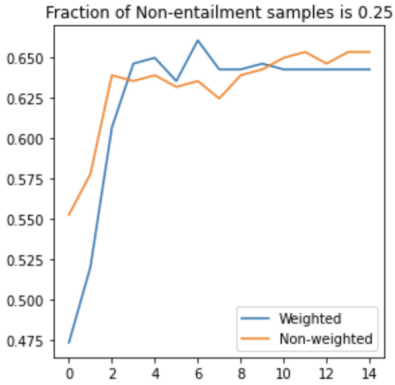
2. Non-Pretrained BERT model

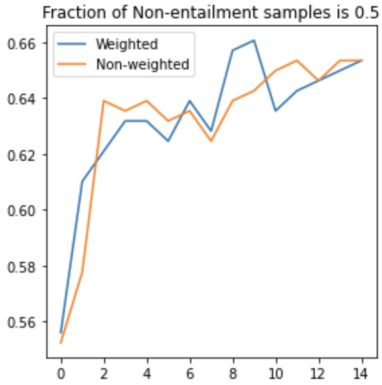
For pretrained BERT models the validation accuracy of non-weighted loss function is lower when there is high data imbalance (i.e. the fraction of class 1 or non_entailement is 0.1 or 0.25) for the initial epochs. This difference in the validation accuracy score reduces as the fraction of class 1 or non_entailement data increases in the training. This observation confirms with our hypothesis “weighted loss function gives better performance than the unweighted loss function with higher accuracy”. But we can see that the difference in the model performance is not much. Also we can see that the validation loss for the weighted loss function is lower than the unweighted loss function in initial epochs thus confirming our hypothesis “weighted loss function gives faster convergence than the unweighted loss function”.

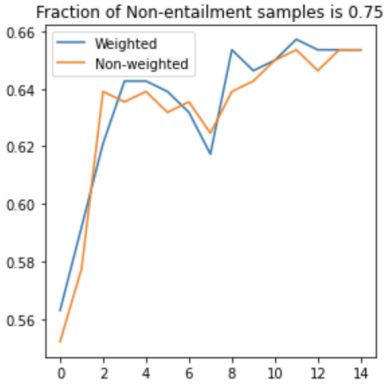
This might be because from the accuracy score we can see that the BERT model does not give high accuracy score with this task and hence we can observe this minor differences in the validation accuracy.

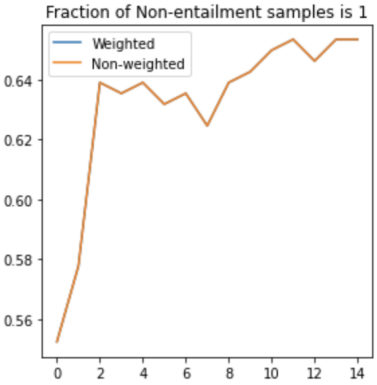
a. Validation Accuracy



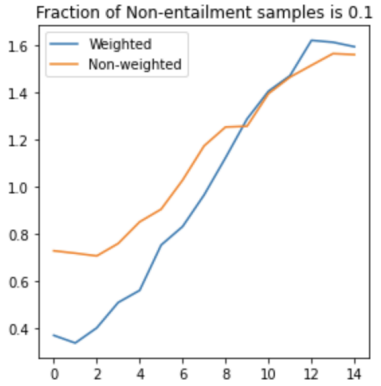


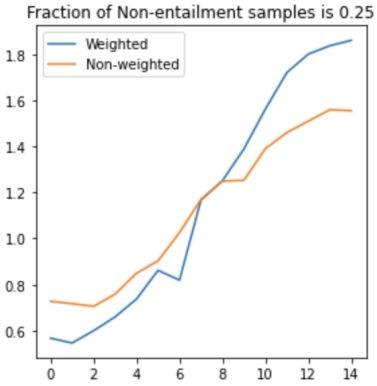


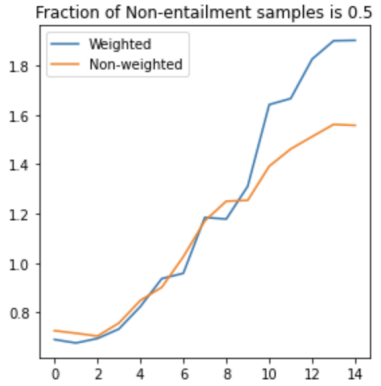


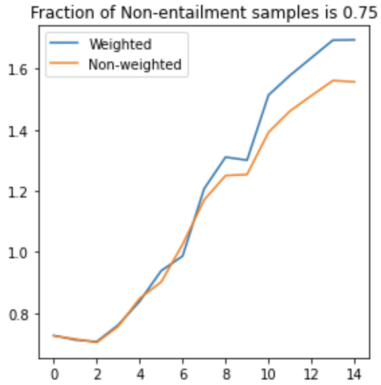


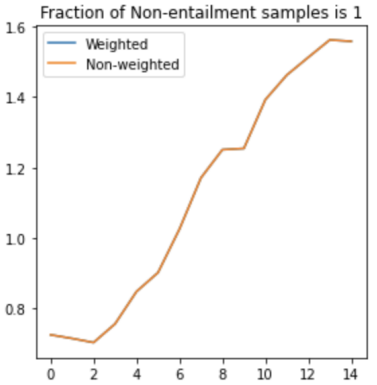
b. Validation Loss









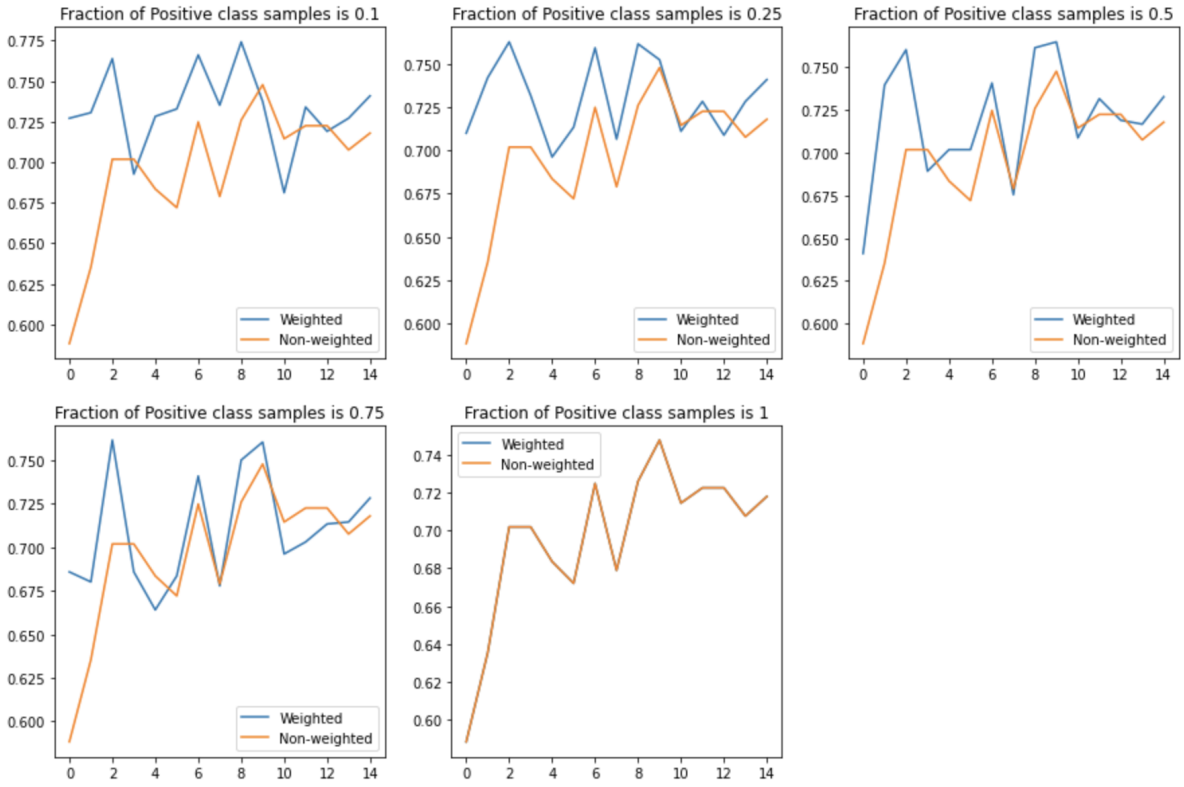


SST-2

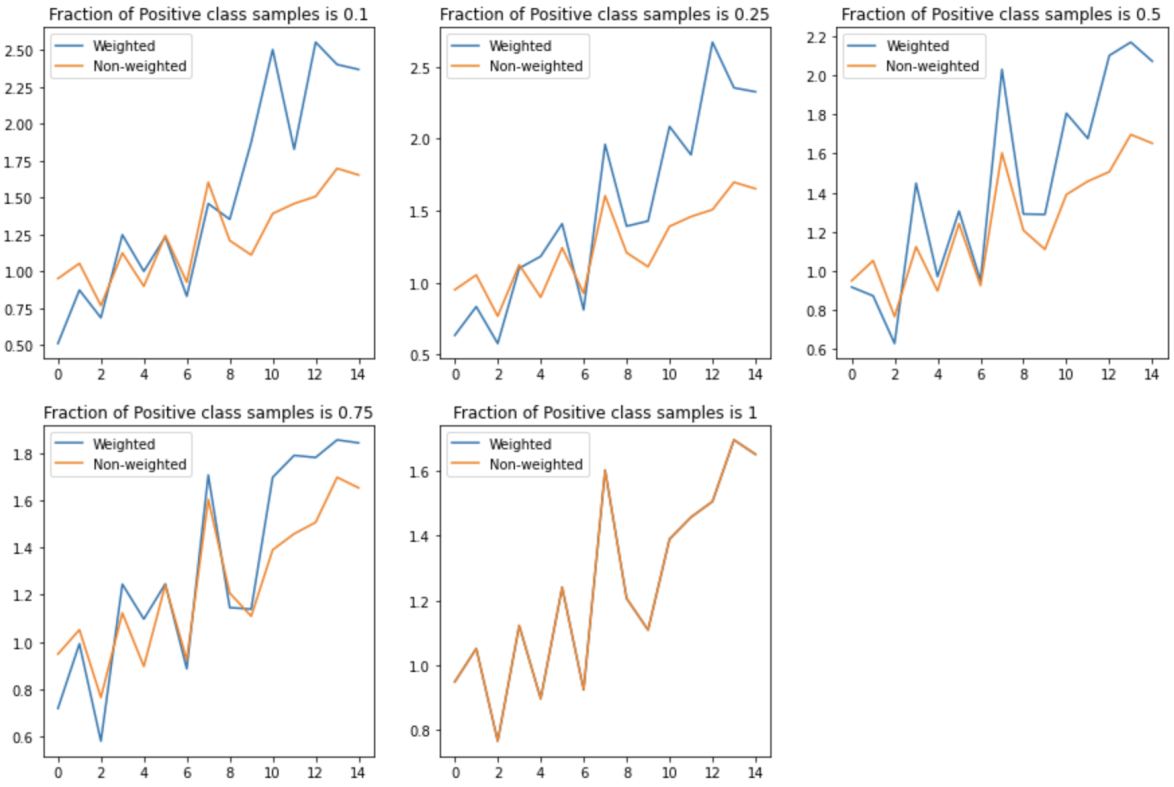
1. Non-Pretrained BERT model

For non-pretrained BERT model the validation accuracy of non-weighted loss function is lower when there is high data imbalance (i.e. the fraction of class 1 or positive is 0.1 or 0.25) for the majority of the epochs. This difference in the validation accuracy score reduces as the fraction of class 1 or positive data increases in the training. This observation confirms with our hypothesis “weighted loss function gives better performance than the unweighted loss function with higher accuracy”. Also we can see that the validation loss for the weighted loss function is lower or remains similar to the unweighted loss function in initial epochs.

a. Validation Accuracy



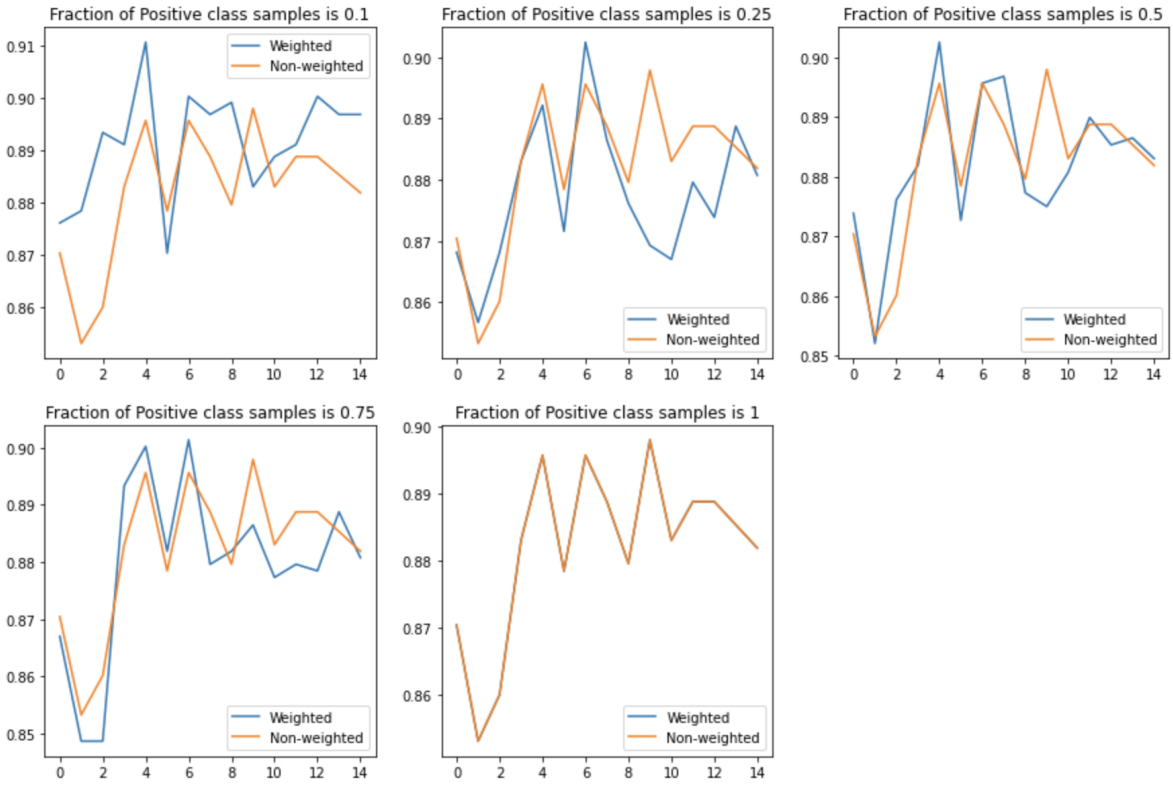
b. Validation Loss



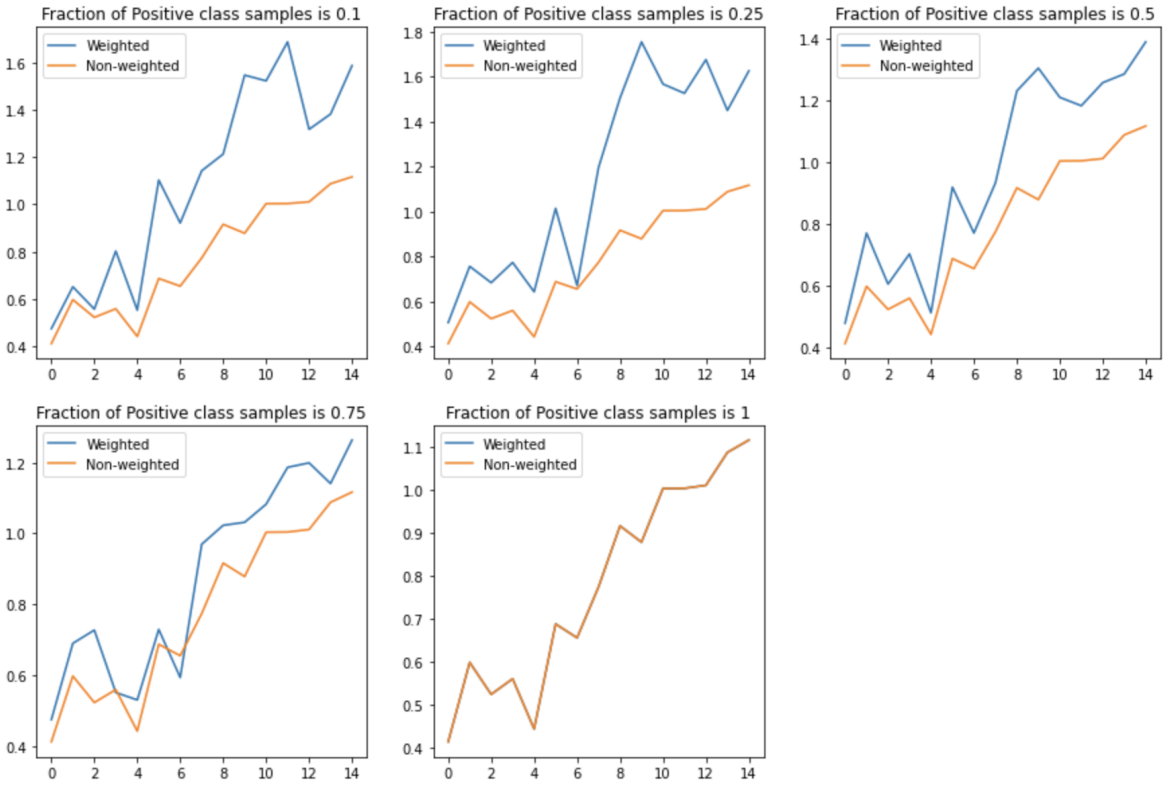
2. Pretrained BERT model

For pretrained BERT model the validation accuracy of non-weighted loss function is lower when there is high data imbalance (i.e. the fraction of class 1 or positive is 0.1 or 0.25) for the initial epochs. This difference in the validation accuracy score reduces as the fraction of class 1 or positive data increases in the training. Also we can see that the validation accuracy score attained in the initial epochs of highly imbalanced data is higher as compared to when the data is balanced.This observation confirms with our hypothesis “weighted loss function gives better performance than the unweighted loss function with higher accuracy”. Also we can see that the validation loss for the weighted loss function is higher than the unweighted loss function in initial epochs. This observation also deviates from our hypothesis “weighted loss function gives faster convergence than the unweighted loss function”.

a. Validation Accuracy



b. Validation Loss



Conclusions

From the experimentation results for the text data we observe that in most of the cases our hypothesis that the weighted loss function gives better performance with higher accuracy is when there is a high class imbalance holds true. On running the experiments for different imbalanced datasets, we observed that more the imbalanced in data, better the effect of the ipw estimator. However, we do not observe faster convergence with the weighted loss functions in the majority of cases. Reasoning for this remains to be explored further.

Bibliography

Byrd, Jonathon, and Zachary C. Lipton. *What is the Effect of Importance Weighting in Deep Learning*. International Conference on Machine Learning}, 2019. *What is the Effect of Importance Weighting in Deep Learning*, https://arxiv.org/pdf/1812.03372.pdf. Accessed 15 05 2021.

Devlin, Jacob, et al. *BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding*. CoRR, 2018, https://arxiv.org/abs/1810.04805. Accessed 15 05 2021.

Dolan, William B., and Chris Brockett. “MRPC Dataset.” *MRPC Dataset | DeepAI*, 2005, https://deepai.org/dataset/mrpc. Accessed 31 03 2021.

Krizhevsky, Alex. “CIFAR-10 and CIFAR-100 Dataset.” *The CIFAR-10 dataset*, 2009, https://www.cs.toronto.edu/~kriz/cifar.html. Accessed 31 03 2021.

Mamaev, Alexander. “Flower Recognition | Kaggle.” *Kaggle*, Kaggle, 2017, https://www.kaggle.com/alxmamaev/flowers-recognition. Accessed 31 03 2021.

Reddi, Sashank, et al. *Doubly robust covariate shift correction*. AAAI Conference on Artificial Intelligence, 2015.

Socher, Richard, et al. *Recursive deep models for semantic compositionality over a sentiment treebank*. Empirical Methods in Natural Language Processing, 2013, https://www.aclweb.org/anthology/D13-1170.pdf. Accessed 15 05 2021.

Wang, Alex, et al. *GLUE: A multi-task benchmark and analysis platform for natural language understanding*. arXiv preprint arXiv:1804.07461, 2018, https://openreview.net/pdf?id=rJ4km2R5t7. Accessed 15 05 2021.

Xu, Da, et al. “Understanding the role of importance weighting for deep learning.” *International Conference on Learning Representations*, 2021, https://openreview.net/forum?id=_WnwtieRHxM.

nal Conference on Machine Learning, in *Proceedings of Machine Learning Research* 97:872-881 Available from http://proceedings.mlr.press/v97/byrd19a.html.