Adaptive weight assignment scheme for multi-task learning

Aminul Huq[1], Mst. Tasnim Pervin[2]
[1]Department of Computer Science and Engineering, Brac University, Dhaka, Bangladesh
[2]Department of Computer Science and Technology, Tsinghua University, Beijing, China

Article Info	ABSTRACT
	Deep learning based models are used regularly in every applications nowadays. Generally we train a single model on a single task. However, we can train multiple tasks on a single model under multi-task learning (MTL) settings. This provides us many benefits like lesser training time, training a single model for multiple tasks, reducing overfitting, and improving performances. To train a model in multi-task learning settings we need to sum the loss values from different tasks. In vanilla multi-task learning settings we assign equal weights but since not all tasks are of similar difficulty we need to allocate more weight to tasks which are more difficult. Also improper weight assignment reduces the performance of the model. We propose a simple weight assignment scheme in this paper which improves the performance of the model and puts more emphasis on difficult tasks. We tested our methods performance on both image and textual data and also compared performance against two popular weight assignment methods. Empirical results suggest that our proposed method achieves better results compared to other popular methods.

Corresponding Author:

Aminul Huq
Department of Computer Science and Engineering,Brac University
66 Mohakhali, Dhaka-1212, Bangladesh
Email: aminul.huq@bracu.ac.bd

1. INTRODUCTION

From the beginning of the last decade deep learning methods has been used vastly in various applications. The reach of it has exceeded tremendously not only in the field of computer science but also in electrical engineering, civil engineering, mechanical engineering and other fields as well. It is due to the fact that deep neural networks (DNN), have achieved human level competence in various applications like image classification [1], question answering [2], lip reading [3], and video games [4]. Deep neural networks have the capability to find out complex and hidden features of the input data without any assistance. Previously these models were depended on hand crafted features [5]-[10].

Human beings have the capability to perform multiple tasks simultaneously without harming performance of any tasks. Humans do this regularly and are able to decide which tasks can be done at the same time. That is why in recent years a lot of focus have been put into multi-task learning using deep neural network methods. Generally, a single model is devoted to performing a single task. However, performing multiple tasks increases the performance of the model, reduces training time and overfitting [11]. Often we find small insufficient datasets for individual tasks but if the tasks are related somehow then we can use this shared information and build a large enough dataset which will reduce this problem. Currently in the field of mult-task learning, several research work is going on to create new deep neural network architectures for multi-task learning setting [12],[13], deciding which tasks should be learned together [14] and how to assign weights to the loss values

[15],[16]. In this research work we focus on creating a dynamic weight assignment technique which will assign different weights to the loss values in each epoch during training. In our research work, we propose a new method for assigning weights to all loss values and test it against two datasets which are used in both image and text domain. The contributions of our research work are: i) We propose an intuitive loss weighting scheme for multi-task learning; ii) We tested our method against both image and text domain by using two different dataset. We did this to ensure that our method performs well across all domains; and iii) We compared our method against two popular weight assigning schemes for comparing the performance of our method

2.   RESEARCH METHOD

In this section we will provide a discussion about previous research work performed in this field. Next, we will provide our proposed method.

2.1.   Literature review

One of the earliest papers on multi-task learning is provided by R. Caruana [11]. In the manuscript, the author explored the idea of multi-task learning and showed it's effectiveness under different datasets. The author also explained how multi-task learning works and how it can be used in backpropagation. To train a deep neural network based on multi-task learning setting we need to consider which layers of network are shared among all the tasks and which layers are used for individual tasks. Previously, most of the research work has been focused on the concept of hard parameter sharing concept [17]-[19]. In this scenario, the user defines the shareable layers up to a particular point after which all layers are assigned per each task. There is also the concept of soft-parameter sharing where a single column exists for all the tasks in the network. A special mechanism is designed to share the parameter across all the network. Popular approaches for this method is cross-stitch [13] and sluich [20]. A new approach named Ada-share has been proposed recently where the model learns dynamically which layers to share for all tasks and which layers to be used for single tasks [14]. The authors also proposed a new loss function which ensures the compactness of the model as well as the performance of it.

Weight assignment is a very crucial task in the field of multi-task learning. Previously weights either had equal values or some hand-tuned value was assigned by the researchers [18],[21],[22]. However in scenarios where a large number of tasks existed for the multi-task learning model to perform, such approaches fall short. A method based on uncertainty was proposed by [15]. Later a revised method of this approach was proposed by [12]. In this paper, the authors improved the previous uncertainty based method by adding a positive regularization term. Dynamic weight average method was proposed by [12]. In this method the authors calculated the relative change in loss values in previous two epochs and used softmax function on these values to get the weights. In the paper, Gong *et al.* [23] performed a comparative study of different weight assigning scheme. However, they didn't study these methods in any domain other than images. Also, the dataset they used had only 2 tasks.

2.2.   Adaptive weight assignment

Our proposed method is simple and it takes into account of the loss value of each task in each epoch. Compared to other methods our method is easy to implement. Generally, in multi-task learning settings to train the model we need to sum up all the loss values with their weights and then perform backpropagation for updating the weights of the model. This summation of losses can be expressed as (1),

$$\sum_{i=1,2,..n} W_i L_i = W_1 L_1 + W_2 L_2 + ... + W_n L_n. \tag{1}$$

here, W corresponds to the weight of the loss and L represents the loss for each task. In vanilla multi-task learning setting all the weights are set to 1. However, we must keep in mind that all the tasks are not the same. Some are more difficult than others so we need to provide more weights on difficult tasks to improve performance of the overall multi-task learning system. That is why we propose algorithm 1.

Our algorithm is based on the simple concept that difficult tasks will have more loss values than the easier ones. So we should put more emphasis or weights on those loss values while assigning less weights to the smaller loss values. What we do is take the sum of the loss values for each tasks and use it to figure out the ratio of how much a single tasks loss value contributes to the total loss. We multiply this value with the total number of tasks. Generally, in vanilla multi-task learning setting all loss values have equal weights 1. So

the total weight is then n for n number of tasks. That is why we multiply our ratios with n. Finally, we use these weights and using (1) compute the total loss for the multi-task learning model. Figure 1 provides a visual representation of the method.

Algorithm 1

Inputs: Loss values $L_1, L_2, .., L_n$, total no. of tasks n
Outputs: Total loss

1: **for** $t = 1, 2, \ldots n$ **do**
2:     $TempLoss$ += L_t
3: **end for**
4: **for** $t = 1, 2, \ldots n$ **do**
5:     $weights_t = L_t/TempLoss$
6:     $TotalLoss$ += $weights_t$ x L_t x n
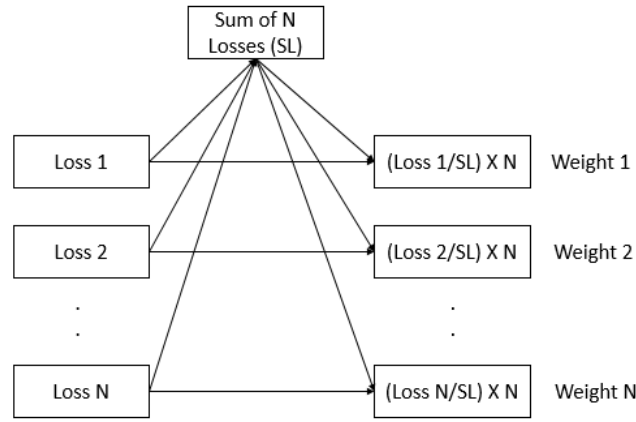7: **end for**

Figure 1. Flow diagram of our proposed method

One of the important things about designing loss weighting schemes is that we need to ensure that these weight calculating methods should not take a lot time because it will increase the training time. Table 1 provides a chart about the time required to execute these schemes including our method. From the table we can see that though our method is not the fastest method to compute weights but it certainly is not the slowest. Also, the time difference between the quickest method and our method is very small.

Table 1. Time required(s) for executing loss weighting schemes on CIFAR-100 and AGNews dataset

	CIFAR-100	AGNews
Re_Uncertainty	0.001	0.0004
DWA	0.0004	0.0002
Ours	0.0006	0.0003

3.   RESULTS AND DISCUSSION

We will discuss about the dataset, experimental setup and results of the experiments in this section.

3.1. Dataset description

We used two different datasets in our experiment. They are CIFAR-100 [24] and AGNews [25]. The formal one is image based and the later one is text based. Since these datasets are designed for single task learning we created artificial tasks for multi-task learning settings. We created 5 different tasks from CIFAR-100 and 2 tasks from AGNews dataset. All the tasks were created based on the original tasks labels and we grouped different labels together to form multiple tasks. The tasks were created to ensure that no class imbalance exists for all tasks.

3.2. Experimental setup

We used two different deep neural network models for our experiment. We used wide resnet-28-10 (WRN) [26] for CIFAR-100 and a custom deep neural network for AGNews dataset. We split the final layer of the WRN model into 5 output layers for CIFAR-100 and 2 output layers for AGNews dataset. we trained WRN model for 100 epochs using SGD optimizer and set the learning rate to 0.001. We also used one cycle learning rate scheduler [27]. In order to train the AGNews dataset we at first tokenize the dataset and create a vocabulary dictionary based on it. Then we perform embedding of the text which is going to be the input of the model. Our custom deep neural network consists of two fully connected layers. We trained this model using SGD optimizer. To ensure the effectiveness of our method, we compared our proposed method against two state-of-the-art methods namely dynamic weight average (DWA) and uncertainty method. We also compared against single task learning and vanilla multi-task setting.

3.3. Experimental results

We will discuss about the performance of our method against two datasets in this section. Tables 2 and 3 represents the results of our overall experiment. In Figure 2 We have plotted test loss curves for the dataset as shown in Figures 2(a) and 2(b) CIFAR 100 and AGNew. In Table 2, we have the results on running experiments on CIFAR-100 dataset which is an image dataset. At the beginning we have results for all the five tasks in a single task learning settings. That is five different models were trained to get the results of these five tasks. Next under multi-task learning setting we trained four methods for these tasks. In vanilla multi-task learning we have assigned equal weights to each task for each epoch. Other methods Uncertainty, DWA and our method updates weights in each epoch. From this table we can see our proposed method out performs other methods in three out of five tasks. Also our method achieved second best performance in the rest of the two tasks. We can see that multi-task learning models performed better than STL models and also we needed to train only one single model for all five of these tasks.

Table 2. Accuracy (%) comparison of different methods and showing best scores (bold) and second best scores (italic)

	2 Class Classification	3 Class Classification	4 Class Classification	5 Class Classification	100 Class Classification
STL	74.52	*75.70*	*74.02*	**72.81**	**76.56**
MTL - Vanilla	79.97	74.36	70.97	67.95	60.23
MTL - Uncertainty	69.47	59.52	55.42	50.21	34.91
MTL - DWA	*80.33*	74.57	71.37	68.41	60.40
MTL - Ours	**81.68**	**77.01**	**74.41**	*72.07*	*66.81*

Table 3. Accuracy (%) comparison of different methods on AGNews dataset and showing best scores (bold) and second best scores (italic)

	2 Class Classification	4 Class Classification
STL	84.00	79.13
MTL - Vanilla	**86.57**	*80.11*
MTL - Uncertainty	84.56	75.94
MTL - DWA	85.86	79.77
MTL - Ours	*86.02*	**81.18**

We evaluate our methods performance on AGNews dataset which contains textual data. We have two tasks and at the beginning we train two individual models for these two tasks. After that we train four multi-task learning models with different weight assignment schemes. We can observe from the table that our proposed method performs well under one task and achieves second best score in the other one. Compared to other popular methods we can see that our proposed method is performing much better. If we look closely at the values we will see that other methods fail to achieve the best results. In some cases these approaches even fail to attain better performance than single task learning approach. We believe this is due to the fact the model architecture has a big impact on the performance of multi-task learning settings. In our experiment we focused on uniform deep neural network architecture for evaluation but some tasks might need a few extra convolutional or fully connected layers. If we put further emphasis on the deep neural network architecture then the performance of our proposed method would definitely be better in both tasks. We believe that a simpler

approach should be taken while assigning weights. As this step is performed in each iteration, too much parameterized and complex approach mind hinder the performance of the model and increase time complexity.
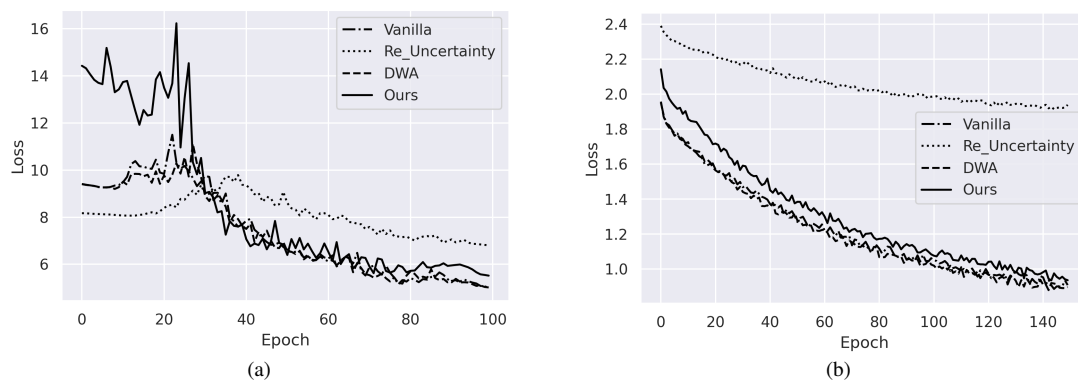

Figure 2. Loss vs epoch curve (a) CIFAR-100 and (b) AGNews

4. CONCLUSION

Understanding and properly executing different hyper-parameters is extremely crucial in training a deep neural network model for the best results. Multi-task learning settings have the upper-hand on single task learning when it comes to amount of data needed, time to train the model, reducing overfitting and increasing model performance. In multi-task learning settings since not all tasks are of equal difficulties assigning weight to the loss values is important to put more emphasis on difficult task. In this paper, we propose a new weight assignment scheme which aids in improving the performance of the multi-task learning model. Our proposed method out-performs other state-of-the-art weight assigning schemes in both image and textual domain and boosts the performance of the model.

REFERENCES

[1] J. Deng, W. Dong, R. Socher, L.-J. Li, Kai Li, and Li Fei-Fei, "ImageNet: A large-scale hierarchical image database," in *2009 IEEE Conference on Computer Vision and Pattern Recognition*, Jun. 2010, pp. 248–255, doi: 10.1109/cvpr.2009.5206848.

[2] P. Rajpurkar, J. Zhang, K. Lopyrev, and P. Liang, "SQuad: 100,000+ questions for machine comprehension of text," in *Conf. on Empirical Methods in Natural Language Processing*, 2016, pp. 2383–2392, doi: 10.18653/v1/d16-1264.

[3] Y. M. Assael, B. Shillingford, S. Whiteson, and N. de Freitas, "LipNet: End-to-End Sentence-level Lipreading," *arXiv preprint arXiv:1611.01599*, Nov 2016.

[4] J. X. Chen, "The evolution of computing: AlphaGo," *Computing in Science Engineering*, vol. 18, no. 4, pp. 4–7, Jul. 2016, doi: 10.1109/MCSE.2016.74.

[5] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," *Proc. IEEE Comp. Society Conf. on Computer Vision and Pattern Recognition, CVPR*, vol. I, 2005, pp. 886–893, doi: 10.1109/CVPR.2005.177.

[6] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International journal of computer vision*, vol. 60, no. 2, pp. 91–110, Nov. 2004, doi: 10.1023/B:VISI.0000029664.99615.94.

[7] R. Mehrotra, K. R. Namuduri, and N. Ranganathan, "Gabor filter-based edge detection," *Pattern Recognit.*, vol. 25, no. 12, pp. 1479–1494, Dec. 1992, doi: 10.1016/0031-3203(92)90121-X.

[8] A. Khotanzad and Y. H. Hong, "Invariant image recognition by Zernike moments," *IEEE Transactions on pattern analysis and machine intelligence*, vol. 12, no. 5, pp. 489–497, May 1990, doi: 10.1109/34.55109.

[9] M. T. Pervin, S. Afroge, and A. Huq, "A feature fusion based optical character recognition of Bangla characters using support vector machine," in *3rd International Conference on Electrical Information and Communication Technology, EICT 2017*, Dec. 2018, vol. 2018-January, pp. 1–6, doi: 10.1109/EICT.2017.8275138.

[10] A. Huq, S. Afroge, and M. T. Pervin, "Combined zemike moments, binary pixel and histogram of oriented gradients feature extraction technique for recognizing hand written Bangla characters," in *3rd Int, Conference on Electrical Information and Communication Technology*, EICT 2017, vol. 2018, 2018, pp. 1–5, doi: 10.1109/EICT.2017.8275144.

[11] R. Caruana, "Multitask Learning," in textslLearning to Learn, Springer US, 1998, pp. 95–133.

[12] S. Liu, E. Johns, and A. J. Davison, "End-to-end multi-task learning with attention," in *Proc. IEEE Comp. Society Conf. on Comp. Vision and Pattern Recognition*, vol. 2019, 2019, pp. 1871–1880, doi: 10.1109/CVPR.2019.00197.

[13] I. Misra, A. Shrivastava, A. Gupta, and M. Hebert, "Cross-Stitch Networks for Multi-task Learning," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, Jun. 2016, vol. 2016-December, pp. 3994–4003, doi: 10.1109/CVPR.2016.433.

[14] X. Sun, R. Panda, R. Feris, and K. Saenko, "AdaShare: Learning what to share for efficient deep multi-task learning," in *Advances in Neural Information Processing Systems*, 2020, vol. 2020-December, [Online]. Available: https://proceedings.neurips.cc/paper/2020/hash/634841a6831464b64c072c8510c7f35c-Abstract.html.

[15] R. Cipolla, Y. Gal, and A. Kendall, "Multi-task learning using uncertainty to weigh losses for scene geometry and semantics," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, Jun. 2018, pp. 7482–7491, doi: 10.1109/CVPR.2018.00781.

[16] L. Liebel and M. Körner, "Auxiliary Tasks in Multi-task Learning," *arxiv.org/abs/1805.06334*, 2018.

[17] J. Huang, R. Feris, Q. Chen, and S. Yan, "Cross-domain image retrieval with a dual attribute-aware ranking network," in *Proc. of the IEEE Int. Conf. on Comp. Vision*, vol. 2015, 2015, pp. 1062–1070, doi: 10.1109/ICCV.2015.127.

[18] I. Kokkinos, "UberNet: Training a universal convolutional neural network for Low-, Mid-, and high-level vision using diverse datasets and limited memory," in *Proc. 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR*, Jul. 2017, vol. 2017, pp. 5454–5463, doi: 10.1109/CVPR.2017.579.

[19] B. Jou and S. F. Chang, "Deep cross residual learning for multitask visual recognition," in *MM 2016 - Proceedings of the 2016 ACM Multimedia Conference*, Oct. 2016, pp. 999–1007, doi: 10.1145/2964284.2964309.

[20] S. Ruder, J. Bingel, I. Augenstein, and A. Søgaard, "Latent multi-task architecture learning," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, pp. 4822–4829, Jul. 2019, doi: 10.1609/aaai.v33i01.33014822.

[21] D. Eigen and R. Fergus, "Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture," in *Proceedings of the IEEE international conference on computer vision*, vol. 2015, 2015, pp. 2650–2658, doi: 10.1109/ICCV.2015.304.

[22] P. Sermanet, D. Eigen, X. Zhang, M. Mathieu, R. Fergus, and Y. LeCun, "Overfeat: Integrated recognition, localization and detection using convolutional networks," *arxiv.org/abs/1312.6229v4*, 2013.

[23] T. Gong et al., "A comparison of loss weighting strategies for multi task learning in deep neural networks," *IEEE Access*, vol. 7, pp. 141627–141632, 2019, doi: 10.1109/ACCESS.2019.2943604.

[24] A. Krizhevsky and G. Hinton, "Learning multiple layers of features from tiny images.(2009)," *Technical report, University of Toronto*, 2009. [Online]. Available: http://www.cs.toronto.edu/ kriz/cifar.html.

[25] X. Zhang, J. Zhao, and Y. Lecun, "Character-level convolutional networks for text classification," *Advances in neural information processing systems*, vol. 2015-January, pp. 649–657, Sep. 2015.

[26] S. Zagoruyko and N. Komodakis, "Wide residual networks," in *British Machine Vision Conference 2016*, BMVC 2016, 2016, vol. 2016-September, pp. 87.1-87.12, doi: 10.5244/C.30.87.

[27] L. N. Smith and N. Topin, "Super-convergence: very fast training of neural networks using large learning rates," in *Artificial Intelligence and Machine Learning for Multi-Domain Operations App.*, May 2019, doi: 10.1117/12.2520589.

BIOGRAPHIES OF AUTHORS

Aminul Huq ⓘ 🔗 SC Ⓟ is a lecturer at Brac University. He received his Master's in Computer Science and Technology from Tsinghua University in 2021. He completed his Bachelor's from Rajshahi University of Engineering & Technology in 2017. His research interest lies in the field of Multi-task Learning and Adversarial Machine Learning. He could be contacted via email: aminul.huq@bracu.ac.bd.

Mst. Tasnim Pervin ⓘ 🔗 SC Ⓟ has completed her Master's in Computer Science and Technology from Tsinghua University in 2021. Her completed her Bachelor's from Rajshahi University of Engineering & Technology in 2017. Her research interest lies in the field of Medical Image Analysis, Adversarial Machine Learning and Domain Adaptation. She could be contacted via email: pervinmt10@mails.tsinghua.edu.cn.