IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received August 22, 2019, accepted September 20, 2019, date of publication September 25, 2019, date of current version October 10, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2943604

A Comparison of Loss Weighting Strategies for Multi task Learning in Deep Neural Networks

TING GONG, TYLER LEE, CORY STEPHENSON, VENKATA RENDUCHINTALA, SUCHISMITA PADHY, ANTHONY NDIRANGO, GOKCE KESKIN, AND OGUZ H. ELIBOL

Intel AI Lab, Santa Clara, CA 95054, USA

Corresponding author: Ting Gong (ting.gong@intel.com)

ABSTRACT With the success of deep learning in a wide variety of areas, many deep multi-task learning (MTL) models have been proposed claiming improvements in performance obtained by sharing the learned structure across several related tasks. However, the dynamics of multi-task learning in deep neural networks is still not well understood at either the theoretical or experimental level. In particular, the usefulness of different task pairs is not known a priori. Practically, this means that properly combining the losses of different tasks becomes a critical issue in multi-task learning, as different methods may yield different results. In this paper, we benchmarked different multi-task learning approaches using shared trunk with task specific branches architecture across three different MTL datasets. For the first dataset, i.e. Multi-MNIST (Modified National Institute of Standards and Technology database), we thoroughly tested several weighting strategies, including simply adding task-specific cost functions together, dynamic weight average (DWA) and uncertainty weighting methods each with various amounts of training data per-task. We find that multi-task learning typically does not improve performance for a user-defined combination of tasks. Further experiments evaluated on diverse tasks and network architectures on various datasets suggested that multi-task learning requires careful selection of both task pairs and weighting strategies to equal or exceed the performance of single task learning.

INDEX TERMS Dynamic weighting average, multi-MNIST, multi-objective optimization, multi-task learning, uncertainty weighting.

I. INTRODUCTION

The goal of multi-task learning (MTL) is to learn multiple different yet related tasks simultaneously [1]. Multi-task learning has been studied across several fields in machine learning. More recently it has been incorporated into a variety of deep neural network models, addressing problems in the domains of vision [2], speech [3], natural language processing [4], and reinforcement learning [5] [6]. The advantages of MTL are 1) the number of parameters in a multi-task model would be fewer than building multiple models each of which is optimized for their own individual tasks; and 2) more importantly, models trained to accomplish many tasks simultaneously should be able to synergize to uncover the common underlying structure, enabling better single-task performance (STL) with smaller amounts of data per-task [7].

Although MTL has shown impressive results on the generalization performance on many tasks, existing MTL studies mainly adopted manually designed feature sharing or parameter sharing on a problem-by-problem basis [8]. The deep learning community still lacks common understanding or rules about 'what to share' and 'how to share', i.e., concrete ways to share knowledge among tasks [8]. To avoid conflicting gradients among tasks, a prevailing issue in MTL, recently researchers also realized that it is critical to find appropriate weighting strategies for MTL so that the total empirical loss is minimized without the trade-off of learning individual tasks [9]. In this work, we empirically evaluate several recently proposed MTL weighting strategies across several MTL benchmark datasets including the Multi-MNIST dataset, the NYU v2 dataset and the IMDB-WIKI dataset.

The weighting approaches we evaluated include uniform combination of losses from different tasks, dynamic weight average (DWA) [10] and uncertainty weighting methods [11] [12] with various amounts of training data per-task. These controlled scenarios help us to frame the problem of

The associate editor coordinating the review of this manuscript and approving it for publication was Bora Onat.

MTL in a manner which can be studied carefully and we can draw significant insights from it reliably. In this paper, our analysis shows that MTL by combining multiple user-defined tasks does not necessarily establish better performance records than single task learning. In order to leverage the knowledge and information across different tasks, we suggest researchers carefully select related tasks and appropriate loss function weighting schemes to optimize the objective functions in MTL.

Information sharing can be achieved using various formalisms and architectures to perform multi-task learning in deep learning [1]. Most existing approaches to multi-task learning attempt to learn a single, yet non-trivial general-purpose representation which is shared among tasks, while keeping several task-specific output layers separate [13]. Thus, we will focus our discussion using this kind of deep learning architecture with shared lower representation layers (trunk), and then fork into task-specific separate layers (heads) and losses for each task. Additionally, in [14], authors categorized MTL into two sub-categories: homogeneous MTL vs. heterogeneous MTL. In homogeneous MTL, each head performs the same type of task i.e., either classification or regression task [15], while in heterogeneous MTL each head may perform different types of task simultaneously or in a sequential order [16]. We will investigate both scenarios in our experiments.

Our contributions are as follows: a) To the best of our knowledge, this is the first meta-analysis that extensively compares the different weighting approaches combining multiple loss functions in the context of both heterogeneous MTL and homogeneous MTL. b) The key observation is that MTL approaches which enforce shared data representations (by using a shared feature extractor network) could show more efficacy when the training samples are sparse. c) We find that many of the results obtained with any chosen set of tasks, which we refer to user-defined tasks, may not achieve performance gains over single task learning (STL), which calls the attention of the deep learning community for more rigorous theoretical analysis.

The remainder of this paper is organized as follows. Section 2 introduces datasets, experimental design and the weighting approaches we compared in our MTL experiments. In Section 3, details of experiments and our results are presented. Finally, we make our conclusions in Section 4 and discuss future directions in MTL.

II. DATASETS AND RELATED WORK FOR WEIGHTING METHODS IN MTL

A. DATASETS

Until now, many recent deep learning approaches have claimed to observe an improvement in performance by sharing learned structure across related tasks [13], [17], [18]. In this paper we investigate several multi-task learning methods and explore when they are most appropriate. MNIST digit recognition is commonly used to evaluate classification algorithms by casting it as 10 binary classification tasks [19]. Given our motivation, we use Multi-MNIST, an MTL version of the MNIST dataset [15]. We follow the experimental settings in [15] and overlay two randomly selected digits together to form the training sample. Each digit is shifted up to 4 pixels in each direction (left and right) resulting in a 36×36 pixel image. It is worth noting that we also use 60K samples instead of 60M samples as done in [15] to directly apply existing single-task MNIST models to Multi-MNIST.

We also evaluate MTL on two popular large-scale datasets: the NYU v2 dataset for a variety of indoor scenes understanding task [20] and the IMDB-WIKI dataset for age and gender classification [21]. For the NYU v2 dataset, we perform the following three tasks [10]: semantic segmentation, depth estimation and surface normal prediction. We use the standard split which include 795 images for training and 654 images for validation. The NYU v2 dataset contains three losses: 1) pixel-wise cross-entropy loss for semantic segmentation, 2) L1 norm comparing the predicted vs. ground-truth depth for depth estimation, and 3) element-wise dot product at each normalised pixel with the ground-truth map for surface normal. So, these tasks fall into the category of heterogeneous MTL. For the IMDB-WIKI dataset, we conduct our experiments to emulate the paper [22]. We use the IMDB portion of the IMDB-Wiki dataset to carry out homogeneous MTL for age and gender classification. The dataset has 460,723 pictures from 20,284 subjects. Using the same preprocessing criterion as [22], we obtain 194,872 usable photos from 12,909 different subjects. Randomly choosing 2,100 test subjects, we retrieve 31,670 test images, leaving us with 10,909 subjects and 163,202 images for training. We further augment the training set with horizontal and vertical flips.

B. WEIGHTING METHODS FOR DIFFERENT LOSS FUNCTIONS IN MTL

In MTL, one tries to optimize multiple loss functions, necessitating a way to combine these loss functions into a single value, or finding solutions where all the loss functions are optimized simultaneously from the multi-objective optimization perspective [9]. In this paper, we limit our experiments to the dynamic task weighting approaches in MTL. The most straightforward way is uniform weighting: the task-specific losses are simply added together to produce a single scalar loss value. However, dynamic optimization techniques are crucial in MTL for optimizing a set of possibly contrasting losses or gradients [9], [11], [23] since problems with conflicting gradient signals coming from different tasks can degrade model performance. Here we focus on two widely adopted methods recently developed. In [11], Kendall *et al.* proposed an uncertainty-based weighting approach and applied it to convolutional neural networks. Very recently, [12] adapted the regularization term in Kendall's uncertainty-based method, but enforced positive regularization values. The revised uncertainty method produced better results compared to the original uncertainty-based method [12]. A dynamic weight average (DWA) has

TABLE 1. Experiments of STL and MTL with two heads of classification tasks on Multi-MNIST dataset.

Approaches	Full Data Set	25% of the Training Samples	Full Data Set	25% of the Training Samples
Single Task (left or right head only)	97.54%(±0.21)	96.20%(±0.42)	96.18%(±0.53)	94.26%(±0.46)
Vanilla MTL	97.50%(±0.34)	96.01%(±0.47)	96.06%(±0.34)	94.07%(±0.55)
DWA MTL	97.28%(±0.28)	95.78%(±0.41)	95.71%(±0.25)	93.57%(±0.45)
Uncertainty MTL	97.14%(±0.31)	95.21%(±0.60)	94.90%(±0.60)	92.16%(±1.16)
Revised uncertainty MTL	97.40%(±0.34)	95.84%(±0.65)	95.55%(±0.36)	93.21%(±0.75)

The left head results are shown in the second and third columns and the right head results are shown in the fourth and fifth columns.

been proposed by [10]. Similarly to GradNorm [23] which averages gradient norms across all tasks during training, DWA first calculates the rate of loss changes for each task during training. DWA then uses a "softmax" output that converts the logit of the loss change ratio into a weight for each task. We use all of these approaches for a fair comparison across multi-task learning techniques in this paper. All the hyper-parameters used are adopted to the default values mentioned in the original papers.

C. NETWORK ARCHITECTURE

We use the LeNet architecture [24] as the shared trunk architecture in our experiments for Multi-MNIST dataset. We treat all layers except the last two layers as an embedding for shared representation and put two fully-connected layers as the last layers of LeNet for task-specific functions.

We evaluate the NYU v2 dataset built upon SegNet, based on their open source implementation of [10]. SegNet is an encoder-decoder architecture based on VGG16. We adopt the version of the standard MTL in [10], which only splits at the last two convolution layers for the prediction of each specific task using the default parameters in the original paper.

Regarding the IMDB-WIKI dataset, we also base our network architecture on VGG16 used in [22]. Here, the split takes place with two more convolution layers and one fully connected layer as the prediction head for each individual task.

D. EVALUATION METRICS

The performance of multi-MNIST is measured by classification accuracy.

For the NYU v2 dataset, we use the same metrics as [10], i.e., semantic segmentation is evaluated by mean Intersection over Union (mIoU) and Pixel Accuracy (PAcc). Depth estimation is measured by Mean Absolute/Relative Error. Surface normal estimation is evaluated by Mean and Median angle distances of all the pixels, as well as the percentage of pixels that are within the angles of 11°, 22.5°, 30° to the ground-truth.

For the IMDB-WIKI dataset, we use the same measurements as in [22]: Mean Absolute Error (Mean AE) and Median Absolute Error (Median AE) for evaluating the age estimation and classification accuracy for gender estimation.

III. EXPERIMENTS

We developed a basic experiment to run on the aforementioned Multi-MNIST dataset to demonstrate the effect of MTL. First, we conducted experiments to compare MTL with Single-Task Learning (STL). We then evaluated the generalization performance under both homogeneous and heterogeneous MTL. We also considered vanilla MTL framework as well as weighted MTL approaches. After having trained on the full dataset, we reported the experiments we carried out on varying amounts of training data to further validate the MTL performance. We further assessed the MTL model's performance by examining the intermediate states of the last layer of the shared trunk. We used t-SNE [25] to visualize and investigated the embeddings/representations taken from the last layer of the shared trunk.

A. MTL FOR DIGIT RECOGNITION

In the first setup, we used LeNet architecture as the trunk, while each head is tasked to recognize the digit separately for an initial baseline. We performed STL, MTL with uniform loss weights, with uncertainty loss weighting (including its revised version), and with DWA. The results are shown in Table 1. We trained all the models with stochastic gradient descent (SGD) using a learning rate of 0.005 and momentum of 0.9, with a batch size of 128. We trained our MTL using Multi-MNIST 60K samples for 200 epochs. Later, each time when we reduce the training sample size by half we double the training time, and so on. We randomly initialized the networks and recorded the best accuracy during training. We repeated this procedure ten times and the numbers shown in Table 1 are the mean accuracy of 10 runs with the variance.

We tested MTL using the full data set first. In Bayesian theory [26], model uncertainty is a quantitative metric revealing the model's inability to do the prediction due to insufficient training data. In order to show the effectiveness of MTL under high model uncertainty, we cut the training samples to 25% of the original input size. From the results showing in Table 1, even in the case with high model uncertainty with a far fewer training samples (only one quarter of the full dataset), potential benefits of MTL are still not observed, although it is intuitively plausible that MTL pushed the network towards learning a robust representation that generalizes well to different atomic tasks [1]. Additionally, in most cases shown in Table 1, neither of the weighting approaches help. We suspect that in our classification MTL problem, conflicting gradient signals from different heads prevent the trunk fully utilizing the knowledge and information from both heads.

TABLE 2. Experiments of STL and MTL with two heads of heterogeneous tasks on Multi-MNIST dataset. Left head: classification of digit and Right head: digit reconstruction regression task.

Approaches	Full Data Set	25% of the Training Samples	10% of the Training Samples
Single Task (left only)	97.02%(±0.30)	95.49%(±0.45)	93.94%(±0.53)
Vanilla MTL	97.29%(±0.42)	95.71%(±0.47)	94.33%(±0.55)
DWA MTL	96.54%(±0.28)	96.36%(±0.43)	93.52%(±0.76)
Uncertainty MTL	97.53%(±0.28)	96.41%(±0.43)	95.03%(±0.59)
Revised uncertainty MTL	**97.67%**(±0.25)	**96.44%**(±0.43)	**95.04%**(±0.69)

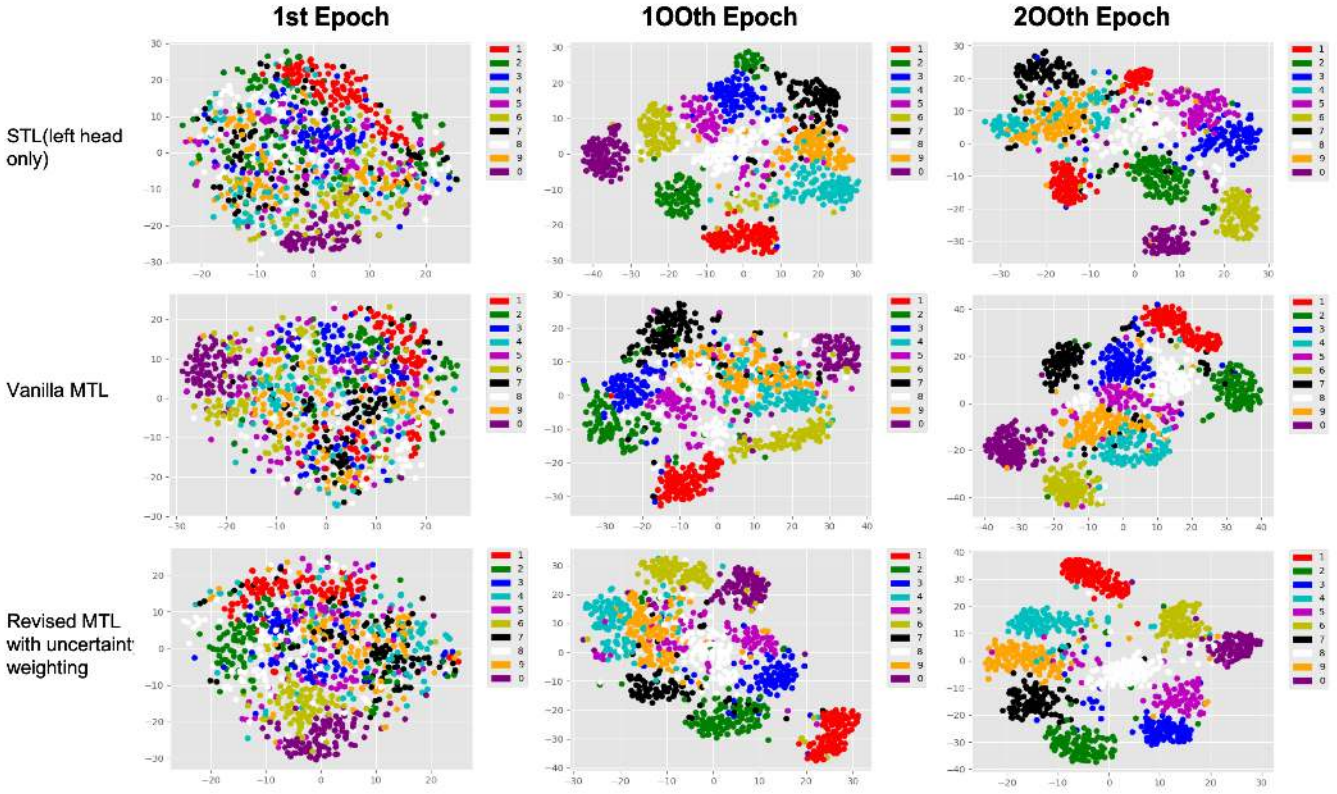

FIGURE 1. tSNE visualization of the output of the embedding layer from the shared trunk. The first column shows the results at the end of 1st epoch, the second column shows the results at the end of 100th epoch and the third column shows the the results at the end of 200th epoch.

B. MTL FOR DIGIT RECOGNITION AND DIGIT RECONSTRUCTION

Secondly, we conducted heterogeneous MTL to further evaluate whether MTL boosts performance by sharing learning structure across classification and regression tasks. We kept the LeNet architecture and left head using the same parameters as the previous setting in Section III A. For the right head, the task was to recover the image of the digit from the shared intermediate representation spaces using a two fully connected layers as decoder. Basically, this head was setup as an autoencoder to recover the digit on the right hand side on the input, with Mean Square Error(MSE) as the loss. We trained our model using SGD with a learning rate of 0.001 and momentum of 0.9. All the other parameters remained the same. Again, we randomly initialized the networks ten times and recorded the best accuracy for each run. Here, to clearly compare with the previous experiment, we focused only on the left head's accuracy.

A summary of the results is given in Table 2, which shows the single-task baseline in the first row and the multi-task results from different weighting methods in the following rows. In this scenario, MTL in most cases achieves higher performance overall compared to the best performance in single-task settings. Especially, when we feed the network with smaller and smaller data sets (i.e., 15K (25% of the whole training samples) and 6K (10% of the whole training samples), the advantages of MTL gradually proved to be more advantageous as the model uncertainty increased. With the full training data set, we achieved the best performance by using the revised uncertainty weighting approach [12] with the increment of performance around 0.65%; and all the other MTL approaches performed slightly better than the single task learning except DWA. When we used 25% of our training samples, we find that the best MTL method is still the revised uncertainty approach by boosting the performance of single task 0.95%. Lastly, we further cut the

TABLE 3. Experimental results on the NYU v2 Dataset. ↑ means the higher the better, while ↓ means the lower the better. Metrics in MTL which are better than STL have been underlined, while the best results are highlighted in bold.

Type	Weighting	Seg		Dept		Surface Normal				
		mIoU↑	Pix Acc↑	Abs Err↓	Rel Err↓	Angle Distance		11.25↑	22.5↑	30↑
						Mean↓	Median↓			
STL	N.A.	16.00 (±0.2147)	**53.56** (±0.4007)	0.6628 (±0.0090)	0.2967(±0.0070)	**29.62** (±0.1613)	**23.40** (±0.1803)	**22.46** (±0.1882)	**45.30** (±0.3034)	**56.73** (±0.3107)
MTL	Equal Weighting	16.13(±0.4048)	52.72(±0.8936)	0.6127(±0.0188)	**0.2694**(±0.0132)	33.23(±0.4690)	27.83(±0.4945)	20.22(±0.6227)	41.66(±0.7484)	53.56(±0.7732)
MTL	DWA	16.15(±0.5527)	52.85 (±1.2209)	**0.6123** (±0.0199)	0.2705 (±0.0126)	33.23 (±0.5825)	27.83(±0.5891)	20.22(±0.5000)	41.66(±0.7948)	53.56(±0.9226)
MTL	Uncertainty	16.53(±0.4680)	53.22(±0.8302)	0.6158 (±0.0171)	0.2702 (±0.0115)	32.34 (±0.3261)	26.54 (±0.3337)	21.39(±0.4191)	43.63(±0.5167)	55.58 (±0.5548)
MTL	Revised Uncertainty	**16.59** (±0.2189)	**53.56** (±0.5102)	0.6132(±0.0165)	0.2713 (±0.0109)	32.29 (±0.3193)	26.61(±0.2918)	21.40 (±0.3467)	43.54(±0.4451)	55.51 (±0.4866

TABLE 4. Experimental results on IMDB-WIKI Dataset. ↑ means the higher the better, while ↓ means the lower the better. Metrics in MTL which are better than STL have been underlined, while the best results are highlighted in bold.

Type	Weighting	Age		Gender
		Mean AE ↓	Median AE ↓	Classification Acc. (%) ↑
STL	N.A.	12.72 (±0.1507)	10.20 (±0.0615)	72.48 (±0.7900)
MTL	Equal Weighting	12.21 (±0.1085)	**9.71** (±0.0958)	72.24 (±0.3300)
MTL	DWA	**12.11** (±0.0390)	9.72 (±0.0919)	72.37 (±0.3900)
MTL	Uncertainty	12.33 (±0.0842)	9.89 (±0.1307)	72.30 (±1.0100)
MTL	Revised Uncertainty	12.25 (±0.0113)	9.82 (±0.0105)	**72.50** (±0.0003)

training samples to 10%. Although we discovered that both of the uncertainty weighting methods became unstable which is also mentioned in [10], vanilla MTL works slightly better than STL, while uncertainty methods help more with revised uncertainty weighting improving the performance by 1.10% comparing to the STL.

C. TSNE VISUALIZATION OF EMBEDDING SPACE

In order to study additional benefits of the heterogeneous multi-task setup, we evaluated the network performance at intermediate optimization states. Figure 1 shows visualizations of the embedding space resulting from Multi-MNIST training. The figure compares tSNE visualizations for STL, vanilla MTL, and the revised uncertainty weighted method. We show the embedding states at training epoch 1, 100 and 200. We can clearly see the advantage of multi-task learning, as embeddings are further separated and clustered, particularly in the final stage of training. For some harder tasks such as distinguishing digits 4 and 9 apart in STL, the representations are mixed in the tSNE projected space. However, when we properly selected the revised weighting method for MTL, the respective embedding of these two digits are well separated. Generally, the separation of 10 classes are more pronounced in MTL compared to single task training, while revised uncertainty shows the most separable pattern for 10 classes. These observations further validate the results seen in our accuracy experiments.

D. PERFORMANCE ON THE NYU V2 DATASET AND THE IMDB-WIKI DATASET

Further, we presented results and analysis of MTL for both the heterogeneous case (NYU v2) and homogeneous case (IMDB-WIKI) using different state-of-the-art deep learning models.

Quantitative results, given in Table 3, show that MTL on NYU v2 dataset offers superior performance on semantic segmentation and depth estimation mostly compared to the STL results. However, MTL does not seem to help surface normal estimation at all, which might need extra investigation. The experimental results also suggest that revised uncertainty weighting outperforms the other weighting approaches, which further validates our observations in the Multi-MNIST experiments.

Experimental results for the IMDB-WIKI dataset are reported in Table 4. The results show that MTL generally outperforms the STL in age estimation. However, MTL barely shows any gains for gender classification. In [22], the authors argue that gender classification is a binary classification problem which has sufficient labels for both classes. Therefore, this task is not expected to benefit much from the other task(s).

IV. CONCLUSION

In this paper, we presented various ways of performing multi-task learning with deep neural networks using the Multi-MNIST dataset and two existing benchmark datasets: NYU v2 and MDB-WIKI daatasets. We evaluated both homogeneous MTL and heterogeneous MTL based on various state-of-art deep learning models, and used different weighting methods (uncertainty weight and DWA approaches) to combine the losses from different tasks.

We conclude that user-defined MTL is not consistently better than STL. When we train our MTL model over an intersection of several tasks, we push the learning algorithm to find a solution on a much smaller yet common area of representation space rather than on a larger area of each individual task. In a lot of the cases, MTL can not improve performance by sharing learned structure across randomly combined tasks when different tasks have conflicting gradient

signals as shown by the results in Table 1. We also highlight that there still is a lack in analytic theories that can quantitatively predict how different tasks can be combined in a multi-task setting. We observe that when tasks are appropriately selected, and proper techniques are used for combining different losses, MTL approach can outperform STL, and the gains are more significant when the amount of training data is small. Several recent papers propose replacing the sum of losses with a weighted sum in MTL, such that all losses are approximately at the same scale and the gradients align together [9] [27]. Here we focused on two recently developed approaches uncertainty loss weighting and DWA. In our current experimental settings, revised uncertainty loss weighting works the best, but it also suffers an instability issue when the size of the training samples is small.

Although we found that there is no principled way of predicting MTL benefit we hope that the results presented in this paper will motivate further investigation by the community into theoretical analyses of multi-task learning.

ACKNOWLEDGMENT

Authors thank our colleagues in Intel AI Lab for useful suggestions and support.

REFERENCES

[1] S. Ruder, "An overview of multi-task learning in deep neural networks," 2017, *arXiv:1706.05098*. [Online]. Available: https://arxiv.org/abs/1706.05098
[2] H. Bilen and A. Vedaldi, "Universal representations: The missing link between faces, text, planktons, and cat breeds," 2017, *arXiv:1701.07275*. [Online]. Available: https://arxiv.org/abs/1701.07275
[3] A. Das, M. Hasegawa-Johnson, and K. Veselý, "Deep auto-encoder based multi-task learning using probabilistic transcriptions," in *Proc. Interspeech*, 2017, pp. 2073–2077. doi: 10.21437/Interspeech.2017-582.
[4] V. Sanh, T. Wolf, and S. Ruder, "A hierarchical multi-task approach for learning embeddings from semantic tasks," Nov. 2018, *arXiv:1811.06031*. [Online]. Available: https://arxiv.org/abs/1811.06031
[5] Y. W. Teh, V. Bapst, W. M. Czarnecki, J. Quan, J. Kirkpatrick, R. Hadsell, N. Heess, and R. Pascanu, "Distral: Robust multitask reinforcement learning," 2017, *arXiv:1707.04175*. [Online]. Available: https://arxiv.org/abs/1707.04175
[6] D. Calandriello, A. Lazaric, and M. Restelli, "Sparse multi-task reinforcement learning," in *Proc. Adv. Neural Inf. Process. Syst.*, Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, Eds. Curran Associates, 2014, pp. 819–827. [Online]. Available: http://papers.nips.cc/paper/5247-sparse-multi-task-reinforcement-learning.pdf
[7] L. Kaiser, A. N. Gomez, N. Shazeer, A. Vaswani, N. Parmar, L. Jones, and J. Uszkoreit, "One model to learn them all," 2017, *arXiv:1706.05137*. [Online]. Available: https://arxiv.org/abs/1706.05137
[8] Y. Zhang and Q. Yang, "A survey on multi-task learning," 2017, *arXiv:1707.08114*. [Online]. Available: https://arxiv.org/abs/1707.08114
[9] O. Sener and V. Koltun, "Multi-task learning as multi-objective optimization," 2018, *arXiv:1810.04650*. [Online]. Available: https://arxiv.org/abs/1810.04650
[10] S. Liu, E. Johns, and A. J. Davison, "End-to-end multi-task learning with attention," 2018, *arXiv:1803.10704*. [Online]. Available: https://arxiv.org/abs/1803.10704
[11] A. Kendall, Y. Gal, and R. Cipolla, "Multi-task learning using uncertainty to weigh losses for scene geometry and semantics," in *Proc. CVPR*, Jun. 2018, pp. 7482–7491.
[12] L. Liebel and M. Körner, "Auxiliary tasks in multi-task learning," 2018, *arXiv:1805.06334*. [Online]. Available: https://arxiv.org/abs/1805.06334
[13] C. Doersch and A. Zisserman, "Multi-task self-supervised visual learning," 2017, *arXiv:1708.07860*. [Online]. Available: https://arxiv.org/abs/1708.07860
[14] Y. Yang and T. M. Hospedales, "Deep multi-task representation learning: A tensor factorisation approach," 2016, *arXiv:1605.06391*. [Online]. Available: https://arxiv.org/abs/1605.06391
[15] S. Sabour, N. Frosst, and G. E. Hinton, "Dynamic routing between capsules," in *Proc. Adv. Neural Inf. Process. Syst.*, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, Eds. Curran Associates, 2017, pp. 3856–3866. [Online]. Available: http://papers.nips.cc/paper/6975-dynamic-routing-between-capsules.pdf
[16] Z. Zhang, P. Luo, C. C. Loy, and X. Tang, "Facial landmark detection by deep multi-task learning," in *Proc. Eur. Conf. Comput. Vis.*, D. Fleet, T. Pajdla, B. Schiele, and T. Tuytelaars, Eds. Cham, Switzerland: Springer, 2014, pp. 94–108.
[17] A. R. Zamir, A. Sax, W. B. Shen, L. J. Guibas, J. Malik, and S. Savarese, "Taskonomy: Disentangling task transfer learning," in *Proc. CVPR*, Jun. 2018, pp. 3712–3722.
[18] B. McCann, N. S. Keskar, C. Xiong, and R. Socher, "The natural language decathlon: Multitask learning as question answering," 2018, *arXiv:1806.08730*. [Online]. Available: https://arxiv.org/abs/1806.08730
[19] A. Kumar and H. Daumé, III, "Learning task grouping and overlap in multi-task learning," in *Proc. Int. Conf. Mach. Learn. (ICML)*, 2012, pp. 1–8. [Online]. Available: http://hal3.name/docs/#daume12gomtl
[20] N. Silberman, D. Hoiem, P. Kohli, and R. Fergus, "Indoor segmentation and support inference from RGBD images," in *Proc. ECCV*, 2012, pp. 746–760.
[21] R. Rothe, R. Timofte, and L. Van Gool, "Deep expectation of real and apparent age from a single image without facial landmarks," *Int. J. Comput. Vis.*, vol. 126, pp. 144–157, Apr. 2018.
[22] Y. Gao, J. Ma, M. Zhao, W. Liu, and A. L. Yuille, "NDDR-CNN: Layerwise feature fusing in multi-task CNNs by neural discriminative dimensionality reduction," 2018, *arXiv:1801.08297*. [Online]. Available: https://arxiv.org/abs/1801.08297
[23] Z. Chen, V. Badrinarayanan, C.-Y. Lee, and A. Rabinovich, "Gradnorm: Gradient normalization for adaptive loss balancing in deep multitask networks," 2017, *arXiv:1711.02257*. [Online]. Available: https://arxiv.org/abs/1711.02257
[24] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proc. IEEE*, vol. 86, no. 11, pp. 2278–2324, Nov. 1998.
[25] L. van der Maaten and G. Hinton, "Visualizing data using t-SNE," *J. Mach. Learn. Res.*, vol. 9, pp. 1533–7928, Nov. 2008.
[26] A. Kendall and Y. Gal, "What uncertainties do we need in Bayesian deep learning for computer vision?" 2017, *arXiv:1703.04977*. [Online]. Available: https://arxiv.org/abs/1703.04977
[27] A. K. Lampinen and S. Ganguli, "An analytic theory of generalization dynamics and transfer learning in deep linear networks," 2018, *arXiv:1809.10374*. [Online]. Available: https://arxiv.org/abs/1809.10374