Supplementary File for Deep Asymmetric Multi-task Feature Learning

Hae Beom Lee [1] **Eunho Yang** [2] **Sung Ju Hwang** [2]

1. Application to Transfer Learning

For this experiment, we use the AWA dataset, which is a standard dataset for transfer learning that provides source/target task class split. The source dataset contains 40 animal classes including *grizzly bear*, *hamster*, *blue whale*, and *tiger*, and the target dataset contains 10 animal classes, including *giant panda*, *rat*, *humpback whale*, and *leopard*. Thus the tasks in two datasets exhibit large degree of relatedness. We train baseline networks and our Deep-AMTFL model on the source dataset, and trained the last fully connected layer of the original network while maintaining all other layers to be fixed, for the classification of the target dataset.

Table 1. Classification error(%) of the baselines and our model on the transfer learning task. Source networks denote types of networks that is trained on the source dataset with 40 classes, and Target accuracy is the accuracy of the softmax classifier on 10 target classes trained on the representations obtained at the layer just below the softmax layer of the source network.

Source Network	Target Accuracy
CNN	5.00
Deep-AMTL	5.00
Deep-AMTFL	**4.33**

2. Experimental Setup

Synthetic dataset experiment All the hyperparameters are found with separate validation sets. For the latent bases models (Go-MTL, AMTFL), we use one hidden layer with six neurons, while other models (STL and AMTL) do not have any hidden layer. The base learning rate is 0.1, and is multiplied by 0.2 every 500 iterations. The batch size is set as 100. The total number of iterations is 2500. RMSProp is used for the latent bases models (Go-MTL, AMTFL), and SGD is used for the other models (STL, AMTL), which has been empirically found to be optimal for each model. Weight decay is set as 0.02. The weights are initialized with gaussian distribution with 0.01 standard deviation. For Go-MTL, the sparsity for $\boldsymbol{L}$ is 0.3 and μ is 0.2. For AMTL, λ and μ are set as 0.3 and 0.0001 each. For AMTFL, the sparsity for $\boldsymbol{L}$ is 0.5, μ is 0.3, α is 0.2, and γ is 0.003.

Real dataset experiment (shallow models) Here we mention a few important settings for the experiments. The base learning rate varying from 10^{-1} to 10^{-4}, and stepwisely decreases when training loss saturates. Batch size also varies from 10^2 to 10^3, which is jointly controlled with learning rate. The number of hidden neurons is set via cross validation, along with other hyperparameters. The weights are initialized with zero-mean gaussian with 0.01 stddev.

Real dataset experiment (deep models) For **MNIST-Imbalanced** dataset, we ran total 200 epochs with batchsize 100. We used the Adam (Kingma & Ba, 2014) optimizer, with the learning rate starts from 10^{-4} and is multiplied by 0.1 after 100 epochs. We set $\lambda = \mu = 10^{-4}$, $\alpha = 0.1$, and $\gamma = 0.01$. For **CUB** dataset, we ran total 400 epochs with batchsize 125. We used SGD optimizer with 0.9 momentum. Learning rate starts from 10^{-2} and is multiplied by 0.1 after 200 and 300 epochs. We set $\lambda = \mu = 10^{-3}$, $\alpha = 1$ and $\gamma = 10^{-3}$. For **AWA-C** dataset, we ran total 300 epochs with batchsize 125. We used SGD optimizer with 0.9 momentum. Learning rate starts from 10^{-2}, and is multiplied by 0.1 at 150 and 250 epochs. We set $\lambda = \mu = 10^{-4}$, $\alpha = 0.1$ and $\gamma = 10^{-4}$. For **ImageNet-Small** dataset, we ran total $40,000$ iterations with batchsize 30. The base learning rate is 10^{-4} and multiplied by 0.1 at every $4,500$ iteration.

3. Other Baselines

In the below table, we show the performances of the two recently proposed multi-task learning models (Yang & Hospedales, 2017; 2016) on two datasets used in the shallow model experiments. The results show that our AMTFL significantly ouperforms those models.

	MNIST	Room
DMTRL	11.9 ± 0.8	47.2 ± 2.9
TNRDMTL	11.0 ± 1.3	49.4 ± 1.4
AMTFL	$\mathbf{8.68 \pm 0.9}$	$\mathbf{40.4 \pm 2.4}$

*Equal contribution [1]UNIST, Ulsan, South Korea [2]KAIST, Daejeon, South Korea. Correspondence to: Hae Beom Lee <hblee@unist.ac.kr>, Eunho Yang <eunhoy@kaist.ac.kr>, Sung Ju Hwang <sjhwang82@kaist.ac.kr>.

Proceedings of the 35^{th} International Conference on Machine Learning, Stockholm, Sweden, PMLR 80, 2018.

References

Kingma, D. P. and Ba, J. Adam: A method for stochastic optimization. In *ICLR*, 2014.

Yang, Y. and Hospedales, T. Deep Multi-task Representation Learning: A Tensor Factorisation Approach. *ICLR*, 2017.

Yang, Y. and Hospedales, T. M. Trace Norm Regularised Deep Multi-Task Learning. *ArXiv e-prints*, June 2016.