Hub-Pathway: Transfer Learning from A Hub of Pre-trained Models

Yang Shu, Zhangjie Cao, Ziyang Zhang, Jianmin Wang, Mingsheng Long (✉)
School of Software, BNRist, Tsinghua University, China

NEURAL INFORMATION PROCESSING SYSTEMS

Summary

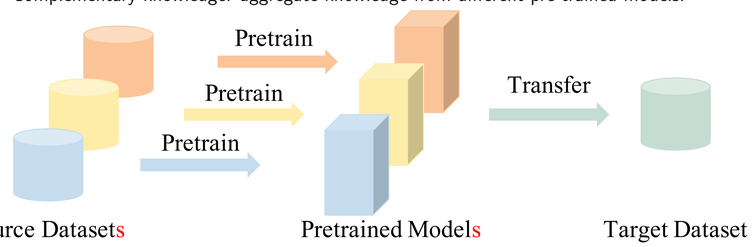
- Explore the problem of transfer learning from a hub of pre-trained models.
- Propose a general **Hub-Pathway** framework that can address different situations where models may be trained from different datasets and learning paradigms and with diverse architectures.
- Promote the **exploration** and **exploitation** of the framework to enhance more effective transfer learning from a model hub.
- Conduct experiments on a variety of model hub transfer learning situations and tasks, including homogeneous and heterogeneous architectures, computer vision and reinforcement learning.

Transfer Learning from a Hub of Pre-trained Models

- **Motivation:** With the development of deep learning methods and large-scale datasets in various fields, and the open-source environment of deep learning community, we now have access to a hub of diverse pre-trained models.

PyTorch TensorFlow Hub Hugging Face Pytorch Image Models

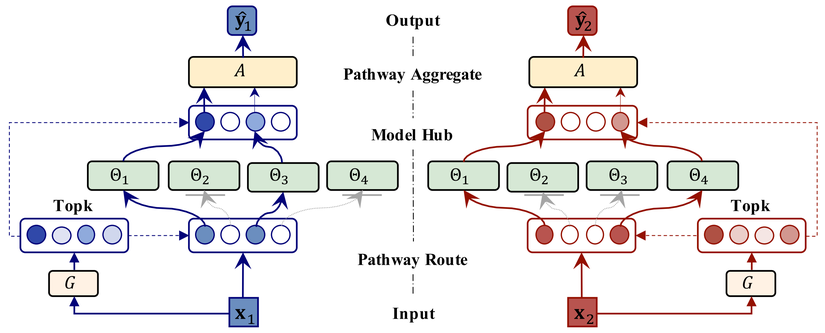
- **Problem Setting:** Transfer learning from a hub of pre-trained models $\{\Theta_1, \Theta_2, \cdots, \Theta_m\}$ to the target task $\mathcal{D} = \{(\mathbf{x}, \mathbf{y})\}$. General situations where models may differ in the pre-training datasets, pretext tasks, learning paradigms and the network architectures.
- **Challenges:**
  - Different relationships: decide which pre-trained models to transfer from.
  - Complementary knowledge: aggregate knowledge from different pre-trained models.



Hub-Pathway Framework

- **General Idea:** Design data-dependent pathways throughout the model hub.
- **Pathway Generator:** Output data-dependent pathway weights $G(\mathbf{x})$ based on the input data $\mathbf{x}$ of the current task. Top-k model activation to ensure efficiency and avoid negative transfer:

$$\bar{G}(\mathbf{x}) = f_{\text{topk}}\left(G(\mathbf{x}), k\right), \text{where } f_{\text{topk}}\left(G(\mathbf{x}), k\right)_i = \begin{cases} G(\mathbf{x})_i & G(\mathbf{x})_i \text{ in the top } k \\ 0 & \text{otherwise.} \end{cases} \quad (1)$$



- **Input-Level Routing:** Assign the pathway route as $\mathbb{I}\left[\bar{G}(\mathbf{x}) > 0\right]$, only pass the input data through the top-k activated models.
- **Output-Level Aggregation:** Compose the knowledge from different models, output final predictions of the model hub: $A\left(\left[\bar{G}(\mathbf{x})_i \cdot \Theta_i(\mathbf{x})\right]_{i=1}^{m}\right)$.
- **Training the Framework:** Learning the pathway routing and the pathway aggregation from the target-task-specific loss:

$$\mathcal{L}_{\text{task}} = \mathbb{E}_{(\mathbf{x},\mathbf{y})\sim\mathcal{D}}\ell\left(A\left(\left[\bar{G}(\mathbf{x})_i \cdot \Theta_i(\mathbf{x})\right]_{i=1}^{m}\right), \mathbf{y}\right) \quad (2)$$

Enhance Exploration of the Model Hub

- **Hub-Collapse:** Over-fit and fall to a local optimum by trivially activating and repeatedly updating with a few specific models. Ignore other potentially useful models and waste the rich knowledge in the hub.
- **Noisy Pathway Generator:** Embody a standard generator subnetwork G_p and a randomized generator subnetwork G_n:

$$G(\mathbf{x}) = \text{Softmax}\left(G_p(\mathbf{x}) + \epsilon \cdot \text{Softplus}(G_n(\mathbf{x}))\right), \epsilon \sim \mathcal{N}(0, 1) \quad (3)$$

- **Pathway Weight Regularization:** Encourage activation of different models and pathways from the dataset point of view. Impose a maximum-entropy regularization on the output of the pathway generator:

$$\mathcal{L}_{\text{explore}} = -\mathcal{H}\left(\mathbb{E}_{(\mathbf{x},\mathbf{y})\sim\mathcal{D}}G(\mathbf{x})\right) \quad (4)$$

Enhance Exploitation of the Model Hub

- **Hub-Underutilization:** Each model contributes only a fraction to hub prediction and optimization. Knowledge is not fully transferred to target.
- **Adaptive Tuning:** Enhance the transfer of knowledge and ensure the performance of the activated models. Tune the pre-trained models $\Theta_i|_{i=1}^{m}$ with the task-specific loss on the specific data that activate them:

$$\mathcal{L}_{\text{exploit}} = \sum_{i=1}^{m}\mathbb{E}_{(\mathbf{x},\mathbf{y})\sim\mathcal{D}}\mathbb{I}\left(\bar{G}(\mathbf{x})_i > 0\right)\ell\left(\Theta_i(\mathbf{x}), \mathbf{y}\right) \quad (5)$$

- **Overall Optimization:** Empowering Hub-Pathway with exploration and exploitation, the final optimization problem becomes:

$$\begin{aligned} &\arg\min_{G,A}\mathcal{L}_{\text{task}} + \lambda \cdot \mathcal{L}_{\text{explore}} \\ &\arg\min_{\Theta_i|_{i=1}^{m}}\mathcal{L}_{\text{task}} + \mathcal{L}_{\text{exploit}} \end{aligned} \quad (6)$$

Experimental Results

- Boost performance on various downstream tasks.

Model	General		Fine-Grained			Specialized		Avg.
	CIFAR	COCO	Aircraft	Cars	Indoors	DMLab	EuroSAT	
ImageNet	81.18	81.97	84.63	89.38	73.69	74.57	98.43	83.41
MoCo	75.31	75.66	83.44	85.38	70.98	75.06	98.82	80.66
MaskRCNN	79.12	81.64	84.76	87.12	73.01	74.73	98.65	82.72
DeepLab	78.76	80.70	84.97	88.03	73.09	74.34	98.54	82.63
Keypoint	76.38	76.53	84.43	86.52	71.35	74.58	98.34	81.16
Ensemble	82.26	82.81	87.02	91.06	73.46	76.01	98.88	84.50
Distill	82.32	82.44	85.00	89.47	73.97	74.57	98.95	83.82
K-Flow	81.56	81.91	85.27	89.22	73.37	75.55	97.99	83.55
ModelSoups	81.32	82.94	85.24	90.32	75.61	74.29	98.65	84.05
Zoo-Tuning-L	83.39	83.50	85.51	89.73	75.12	75.22	**99.12**	84.51
Zoo-Tuning	**83.77**	**84.91**	86.54	90.76	75.39	75.64	**99.12**	85.16
Hub-Pathway	83.31	84.36	**87.52**	**91.72**	**76.91**	**76.47**	**99.12**	**85.63**

- General for model hubs with heterogenous models.

Model	General		Fine-Grained			Specialized		Avg.
	CIFAR	COCO	Aircraft	Cars	Indoors	DMLab	EuroSAT	
MaskRCNN	$79.12_{\pm 0.06}$	$81.64_{\pm 0.39}$	$84.76_{\pm 0.30}$	$87.12_{\pm 0.09}$	$73.01_{\pm 0.45}$	$74.73_{\pm 0.46}$	$98.65_{\pm 0.05}$	82.72
MobileNetV3	$83.14_{\pm 0.10}$	$83.28_{\pm 0.05}$	$80.26_{\pm 0.03}$	$86.37_{\pm 0.61}$	$75.09_{\pm 0.19}$	$70.09_{\pm 0.24}$	$98.95_{\pm 0.11}$	82.45
EffNet-B3	$87.28_{\pm 0.21}$	$86.97_{\pm 0.08}$	$83.99_{\pm 0.09}$	$89.34_{\pm 0.13}$	$78.16_{\pm 0.16}$	$72.69_{\pm 0.27}$	$99.13_{\pm 0.01}$	85.37
Swin-T	$84.37_{\pm 0.12}$	$84.12_{\pm 0.01}$	$80.82_{\pm 0.27}$	$89.10_{\pm 0.09}$	$73.39_{\pm 0.34}$	$72.22_{\pm 0.24}$	$98.69_{\pm 0.05}$	83.24
ConvNeXt-T	$86.96_{\pm 0.10}$	$87.15_{\pm 0.09}$	$84.23_{\pm 0.57}$	$90.67_{\pm 0.04}$	$81.66_{\pm 0.07}$	$73.80_{\pm 0.11}$	$98.65_{\pm 0.04}$	86.16
Ensemble	$87.72_{\pm 0.19}$	$88.04_{\pm 0.07}$	$87.11_{\pm 0.28}$	$92.68_{\pm 0.33}$	$82.79_{\pm 0.26}$	$\mathbf{74.86}_{\pm 0.14}$	$99.23_{\pm 0.01}$	87.49
Distill	$87.33_{\pm 0.16}$	$88.09_{\pm 0.25}$	$85.26_{\pm 0.32}$	$91.39_{\pm 0.19}$	$81.51_{\pm 0.29}$	$74.75_{\pm 0.20}$	$99.24_{\pm 0.02}$	86.80
Hub-Pathway	$\mathbf{89.01}_{\pm 0.06}$	$\mathbf{89.14}_{\pm 0.12}$	$\mathbf{88.12}_{\pm 0.14}$	$\mathbf{92.93}_{\pm 0.20}$	$\mathbf{84.40}_{\pm 0.22}$	$74.80_{\pm 0.23}$	$\mathbf{99.26}_{\pm 0.06}$	**88.24**

- Visualization on learned pathways.

- Complexity analysis, more efficient than ensemble methods.

Model	Acc (%) ↑	Params (M) ↓	FLOPs (G) ↓	Memory (M) ↓	Speed (samples/s) ↑
ImageNet	83.41	23.71	4.11	1905	484.92
Ensemble-J	83.87	118.55	20.55	6397	98.64
Ensemble-I	84.50	118.55	20.55	6397	98.64
Hub-Pathway	85.63	128.43	9.11	3537	240.48

- Overall: Consistently outperform the best single model in the model hub. Effectively use knowledge from diverse pre-trained models, achieve a good balance between performance and efficiency.

School of Software - Tsinghua University - China
Mail: shu-y18@mails.tsinghua.edu.cn