电子科技大学
University of Electronic Science and Technology of China

Modular Neural Network

Wei Han

Data Mining Lab,
Big Data Research Center, UESTC
Email：weihan@std.uestc.edu.cn

Outline

- Motivation & Background
  - Problems of ANN learning
  - Go further to Bio-neurology
  - Modular Neural Network

- Pattern Segregation
  - Theoretical research
  - Methodology
  - Application

Motivation & Background

1.1. Problems of ANN learning

- ‘One-Punch’ Model —— Artificial Neural Network
  - Computer Vision
  - Natural Language Processing
  - Speech Recognition

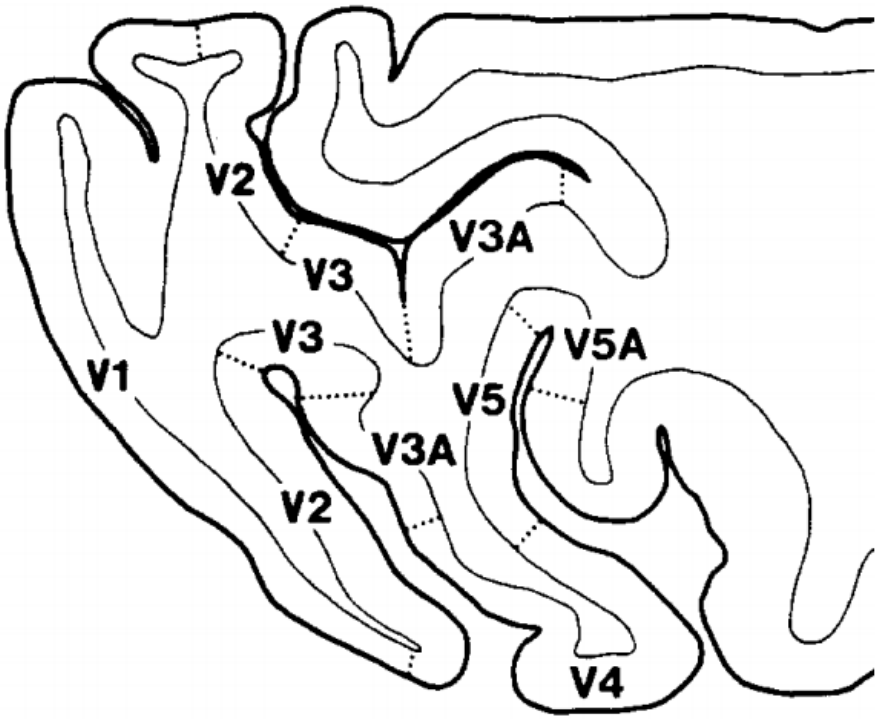
- What’s the problems?
  - Extreme fitting ability but no constraint
  - Chaotic representation

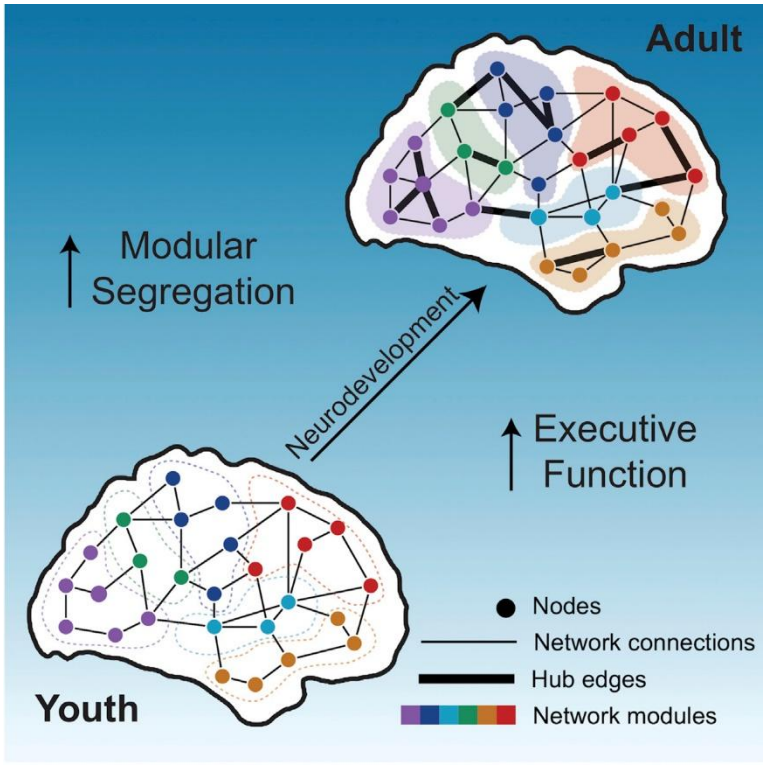
1.1. Problems of ANN learning

- Interpretation
  - Extreme nonlinear & chaotic representation
- Catastrophic Forgetting
  - Unknown previous knowledge storage
- Go further to ANN learning
  - Global -> Local, Coarse -> Fine
  - Learning with pattern arrangement

1.2. Bio-neurology Basis

- Go further to Bio-neurology -> Modularity





1.2. Further Potential

- Interpretation
- Distributed Learning & Super-parallel
- Transfer Learning
- Never cold start
- Lifelong Learning
- Instance Recall vs. Knowledge Interaction

Modular Neural Network

- Key Points of MNN
  - Modules
  - Routing criterion
- Three Kinds of MNN
  - Instance-based MNN
    - Training modules with instances of similar patterns
  - Feature-based MNN
    - Automatically segregating modules by feature constraint
  - Model-based MNN
    - Training physically divided modules

Pattern Segregation

- Similarity of Neural Network Representations Revisited

  - Does the same architecture deep neural network based on different random initialization training learn similar representations?
  - Can a corresponding relationship be established between layers of different neural network architectures?
  - How similar are the representations of the same neural network architecture learned from different data sets?

2.1. Theoretical research

- Background & Motivation
  - **Representational similarities** between neural network models
  - Concept of similarity & dynamic process of training → invariant to orthogonal transformation & isotropic scaling, NOT reversible linear transformation
  - Previous methods: directly compare multivariate features of a sample in two representations
  - Current paper: First measure the similarity between each pair of samples in each representation, then compare the **similarity structure**

2.1. Theoretical research

- Similarity based on dot product

$$\langle \mathrm{vec}(XX^{\mathrm{T}}), \mathrm{vec}(YY^{\mathrm{T}}) \rangle = \mathrm{tr}(XX^{\mathrm{T}}YY^{\mathrm{T}}) = ||Y^{\mathrm{T}}X||_{\mathrm{F}}^2$$

- Hilbert-Schmidt Independence Criterion

$$\frac{1}{(n-1)^2}\mathrm{tr}(XX^{\mathrm{T}}YY^{\mathrm{T}}) = ||\mathrm{cov}(X^{\mathrm{T}}, Y^{\mathrm{T}})||_{\mathrm{F}}^2$$

Let $K_{ij} = k(x_i, x_j), L_{ij} = l(y_i, y_j)$

$$\mathrm{HSIC}(K, L) = \frac{1}{(n-1)^2}\mathrm{tr}(KHLH)$$

$$\mathrm{CKA}(K, L) = \frac{\mathrm{HSIC}(K, L)}{\sqrt{\mathrm{HSIC}(K, K)\mathrm{HSIC}(L, L)}}$$

- Related Works

Similarity Index	Formula	Invariant to		
		Invertible Linear Transform	Orthogonal Transform	Isotropic Scaling
Linear Regression (R^2_{LR})	$\|\|Q_Y^{\mathsf{T}} X\|\|_{\text{F}}^2 / \|\|X\|\|_{\text{F}}^2$	Y Only	✓	✓
CCA (R^2_{CCA})	$\|\|Q_Y^{\mathsf{T}} Q_X\|\|_{\text{F}}^2 / p_1$	✓	✓	✓
CCA ($\bar{\rho}_{\text{CCA}}$)	$\|\|Q_Y^{\mathsf{T}} Q_X\|\|_* / p_1$	✓	✓	✓
SVCCA (R^2_{SVCCA})	$\|\|(U_Y T_Y)^{\mathsf{T}} U_X T_X\|\|_{\text{F}}^2 / \min(\|\|T_X\|\|_{\text{F}}^2, \|\|T_Y\|\|_{\text{F}}^2)$	In a Subspace	✓	✓
SVCCA ($\bar{\rho}_{\text{SVCCA}}$)	$\|\|(U_Y T_Y)^{\mathsf{T}} U_X T_X\|\|_* / \min(\|\|T_X\|\|_{\text{F}}^2, \|\|T_Y\|\|_{\text{F}}^2)$	In a Subspace	✓	✓
PWCCA	$\sum_{i=1}^{p_1} \alpha_i \rho_i / \|\|\alpha\|\|_1, \alpha_i = \sum_j \|\langle \mathbf{h}_i, \mathbf{x}_j \rangle\|$	✗	✗	✓
Linear HSIC	$\|\|Y^{\mathsf{T}} X\|\|_{\text{F}}^2$	✗	✓	✗
Linear CKA	$\|\|Y^{\mathsf{T}} X\|\|_{\text{F}}^2 / (\|\|X^{\mathsf{T}} X\|\|_{\text{F}} \|\|Y^{\mathsf{T}} Y\|\|_{\text{F}})$	✗	✓	✓
RBF CKA	$\text{tr}(KHLH) / \sqrt{\text{tr}(KHKH)\text{tr}(LHLH)}$	✗	✓	✓*

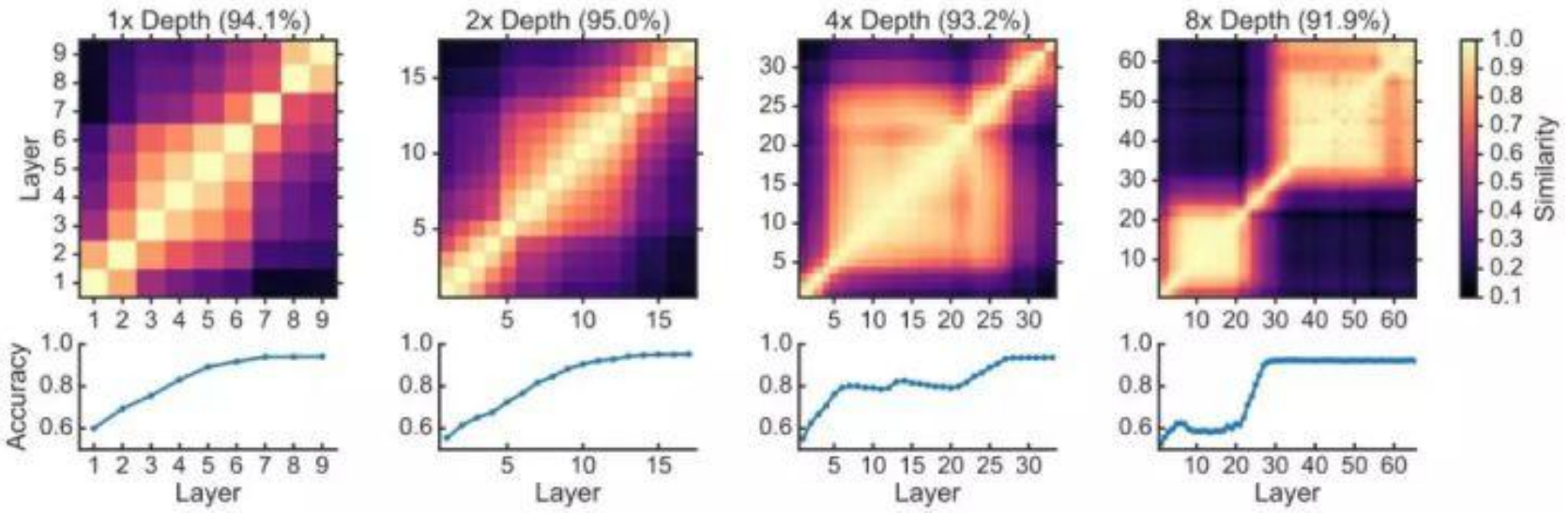
2.1. Theoretical research

- Recognize layers in networks with different initialization

Index	Accuracy
CCA ($\bar{\rho}$)	1.4
CCA (R^2_{CCA})	10.6
SVCCA ($\bar{\rho}$)	9.9
SVCCA (R^2_{CCA})	15.1
PWCCA	11.1
Linear Reg.	45.4
Linear HSIC	22.2
CKA (Linear)	**99.3**
CKA (RBF 0.2)	80.6
CKA (RBF 0.4)	**99.1**
CKA (RBF 0.8)	**99.3**

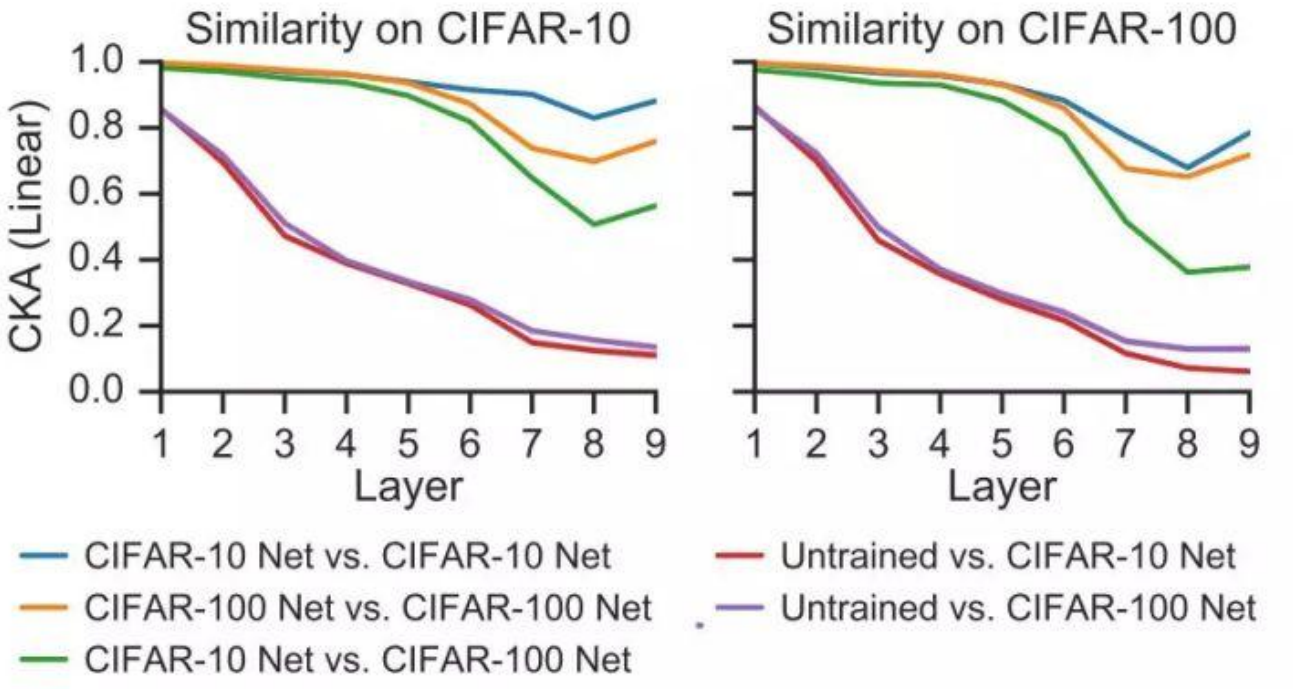
2.1. Theoretical research

- Reveal abnormality in deep neural network

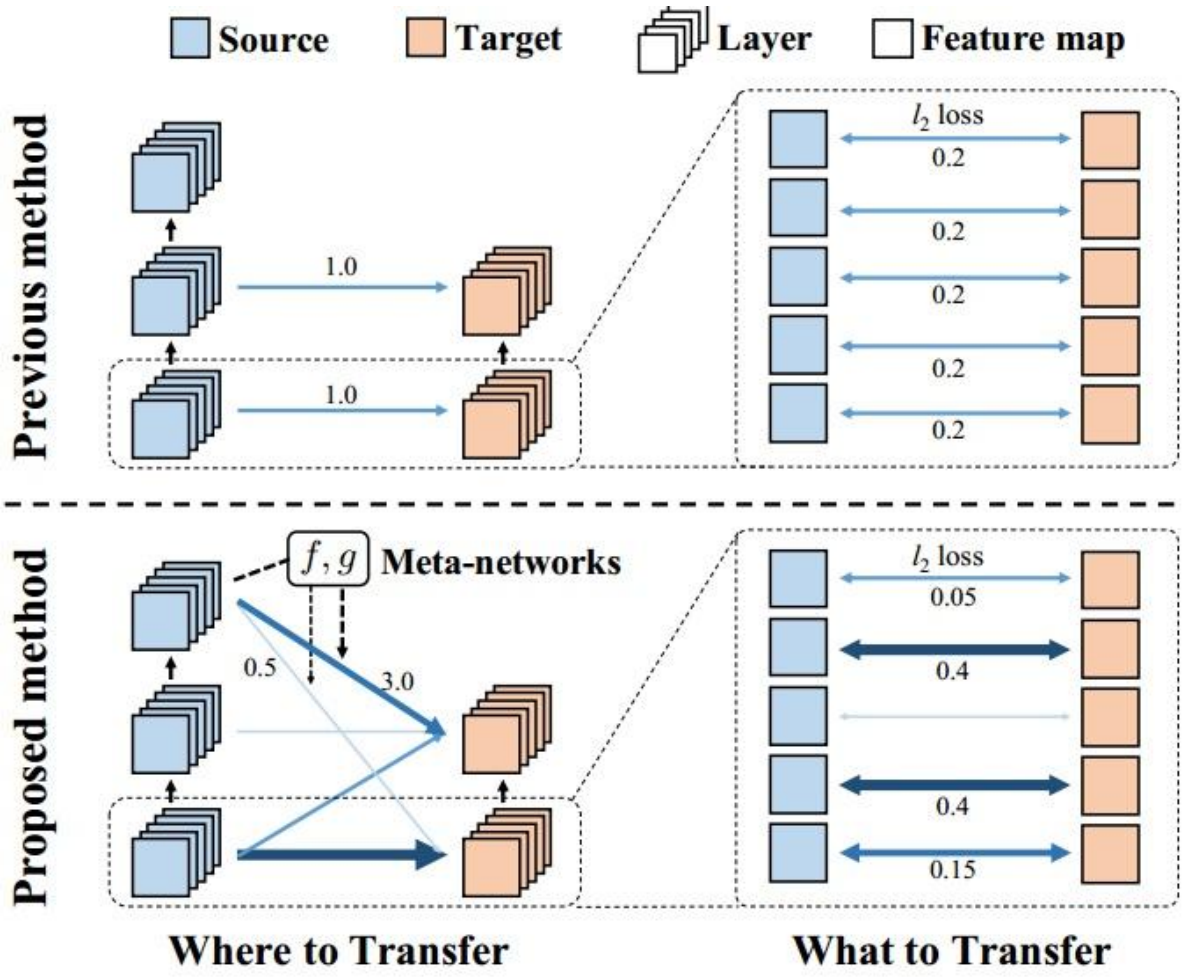


2.1. Theoretical research

- Layer similarity in networks trained with different data set

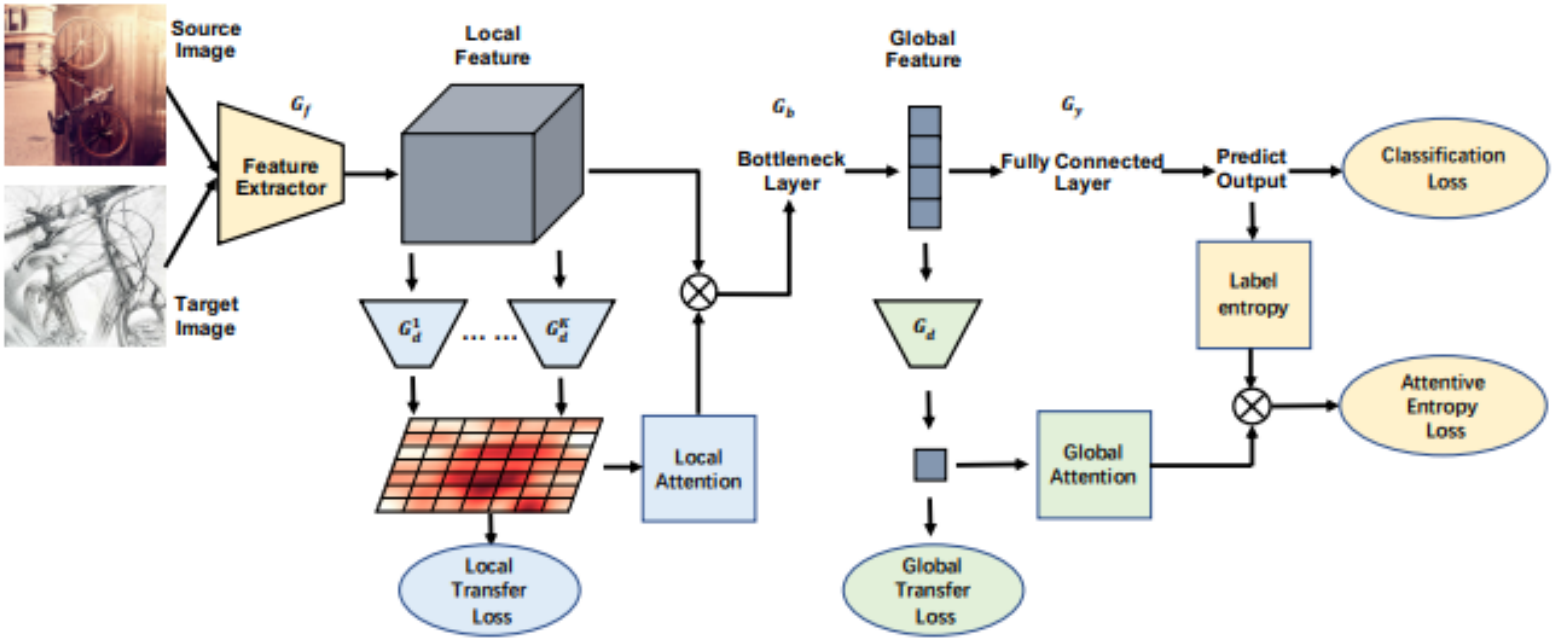


- "Learn What-Where to Transfer"



2.2. Methodology

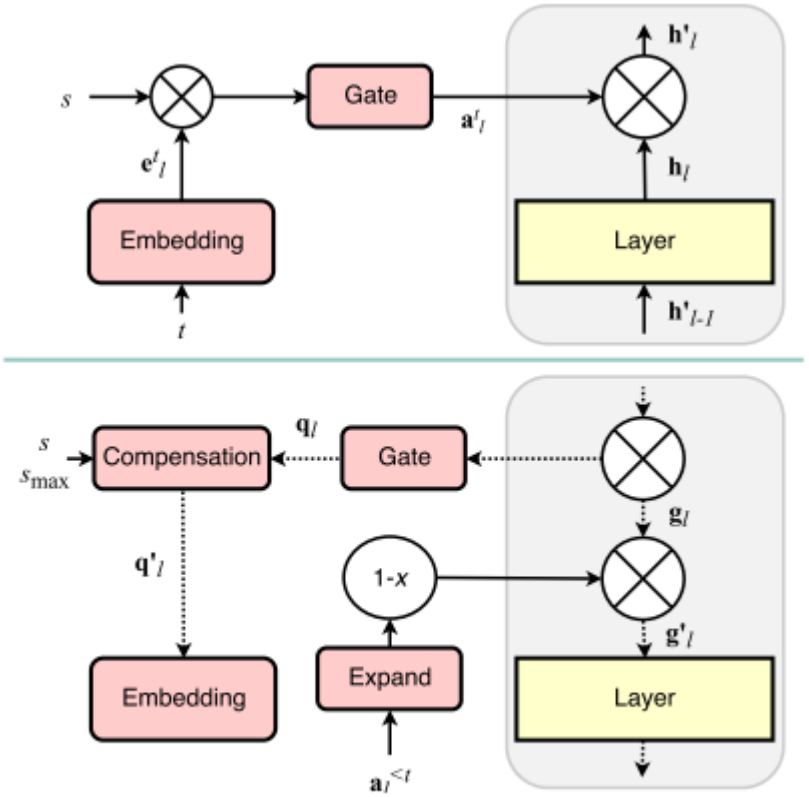
- “Transferable Attention for Domain Adaptation”



2.2. Methodology

- Overcoming catastrophic forgetting with hard attention to the task

$$s = \frac{1}{s_{\max}} + \left(s_{\max} - \frac{1}{s_{\max}}\right)\frac{b-1}{B-1}$$



$$\mathbf{h}'_l = \mathbf{a}^t_l \odot \mathbf{h}_l$$

$$\mathbf{a}^t_l = \sigma\left(s\mathbf{e}^t_l\right)$$

$$g'_{l,ij} = \left[1 - \min\left(a^{\leq t}_{l,i}, a^{\leq t}_{l-1,j}\right)\right] g_{l,ij}$$

Figure 1. Schematic diagram of the proposed approach: forward (top) and backward (bottom) passes.

2.2. Methodology

- Embedding Gradient Compensation

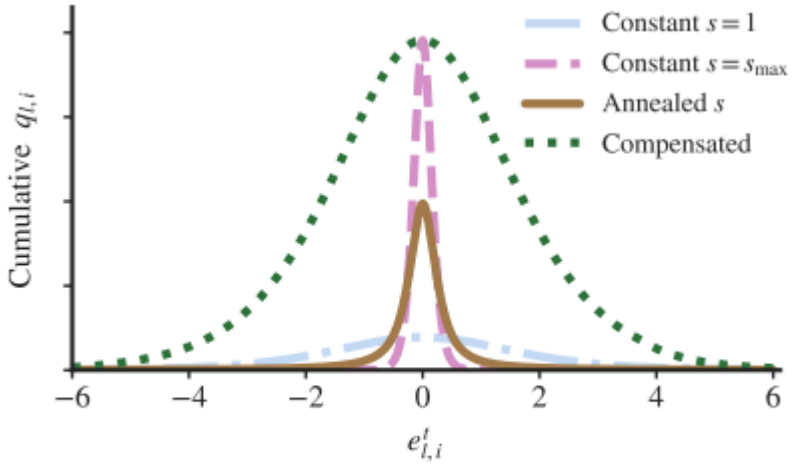


Figure 2. Illustration of the effect that annealing s has on the gradient q of $\mathbf{e}^t$.

$$q'_{l,i} = \frac{s_{\max}\sigma\left(e^t_{l,i}\right)\left[1-\sigma\left(e^t_{l,i}\right)\right]}{s\sigma\left(se^t_{l,i}\right)\left[1-\sigma\left(se^t_{l,i}\right)\right]} q_{l,i}$$

$$q'_{l,i} = \frac{s_{\max}\left[\cosh\left(se^t_{l,i}\right)+1\right]}{s\left[\cosh\left(e^t_{l,i}\right)+1\right]} q_{l,i}$$

- Promoting Low Capacity Usage

$$R\left(\mathsf{A}^t, \mathsf{A}^{<t}\right) = \frac{\sum_{l=1}^{L-1}\sum_{i=1}^{N_l} a^t_{l,i}\left(1-a^{<t}_{l,i}\right)}{\sum_{l=1}^{L-1}\sum_{i=1}^{N_l} 1-a^{<t}_{l,i}}$$

- Results

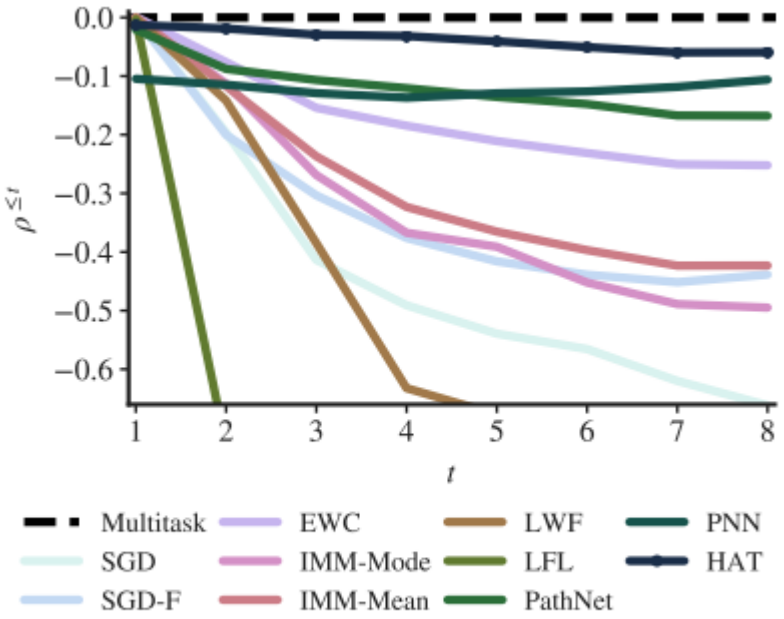


Figure 3. Average forgetting ratio $\rho^{\leq t}$ for the considered approaches (10 runs).

Table 1. Average forgetting ratio after the second ($\rho^{\leq 2}$) and the last ($\rho^{\leq 8}$) task for the considered approaches (10 runs, standard deviation into parenthesis).

APPROACH	$\rho^{\leq 2}$	$\rho^{\leq 8}$
LFL	-0.73 (0.29)	-0.92 (0.08)
LWF	-0.14 (0.13)	-0.80 (0.06)
SGD	-0.20 (0.08)	-0.66 (0.03)
IMM-MODE	-0.11 (0.08)	-0.49 (0.05)
SGD-F	-0.20 (0.15)	-0.44 (0.06)
IMM-MEAN	-0.12 (0.10)	-0.42 (0.04)
EWC	-0.08 (0.06)	-0.25 (0.03)
PATHNET	-0.09 (0.16)	-0.17 (0.23)
PNN	-0.11 (0.10)	-0.11 (0.01)
HAT	**-0.02 (0.03)**	**-0.06 (0.01)**

2.2. Methodology

- Monitoring and Network Pruning

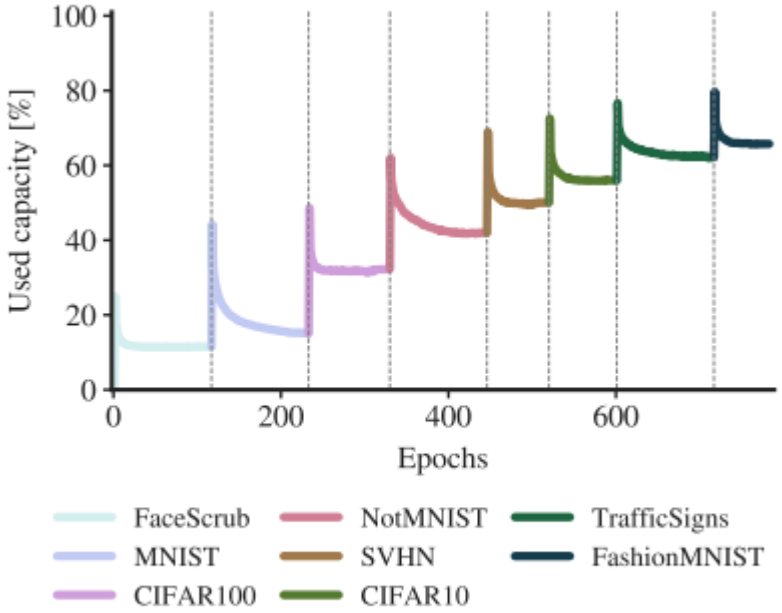


Figure 5. Network capacity usage with sequential task learning (seed 0). Dashed vertical lines correspond to a task switch.

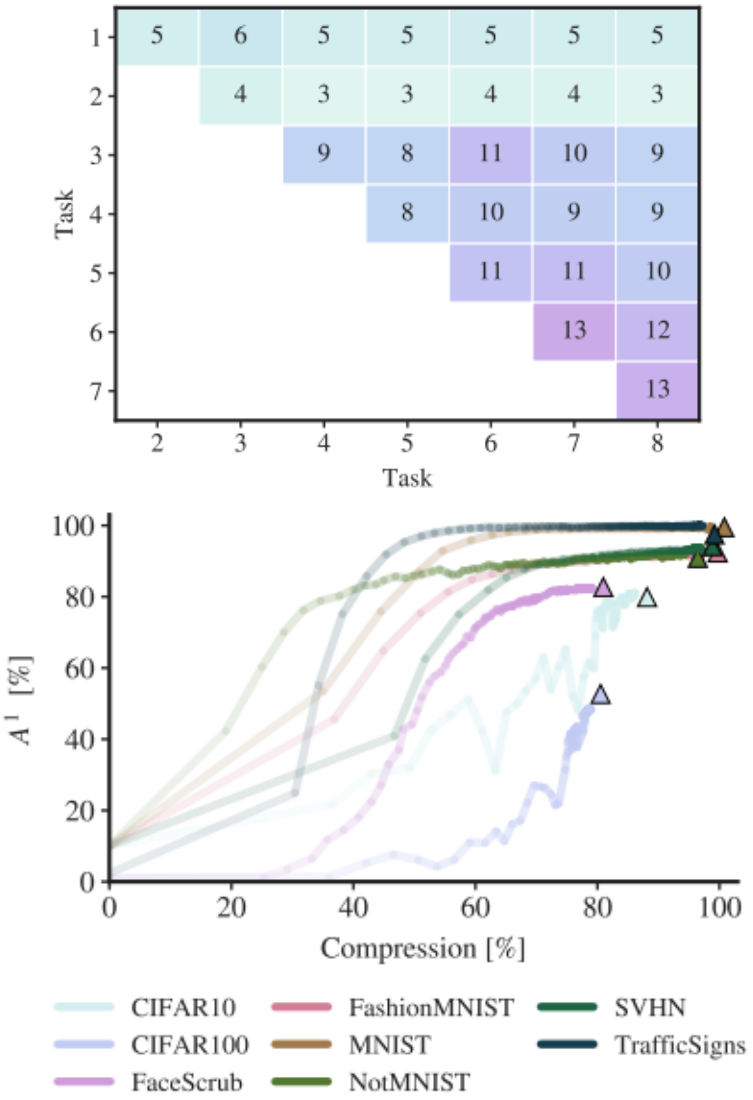
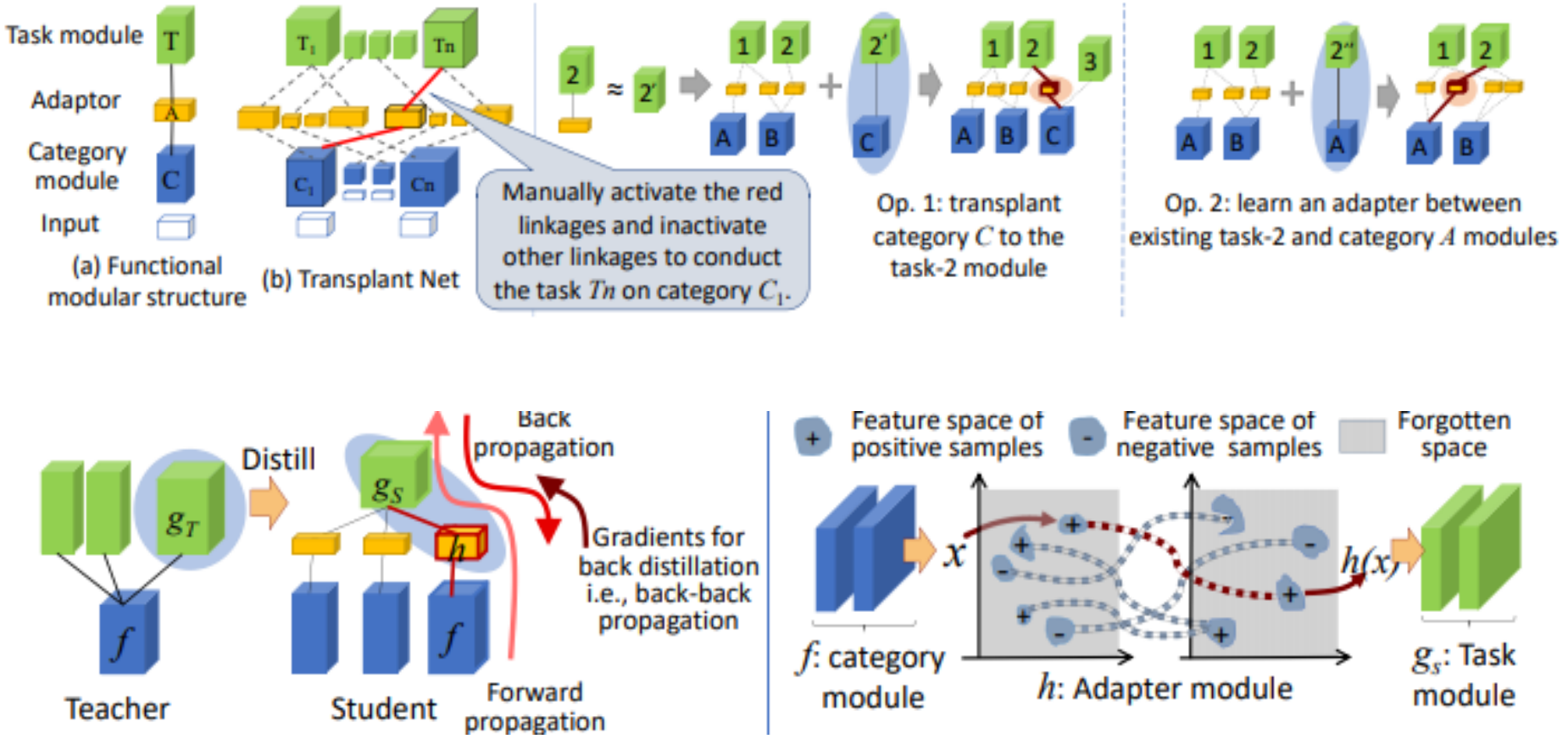


Figure 7. Validation accuracy A^1 as a function of compression percentage. Every dot corresponds to an epoch and triangles match the accuracy of the SGD approach (no compression).

2.2. Methodology

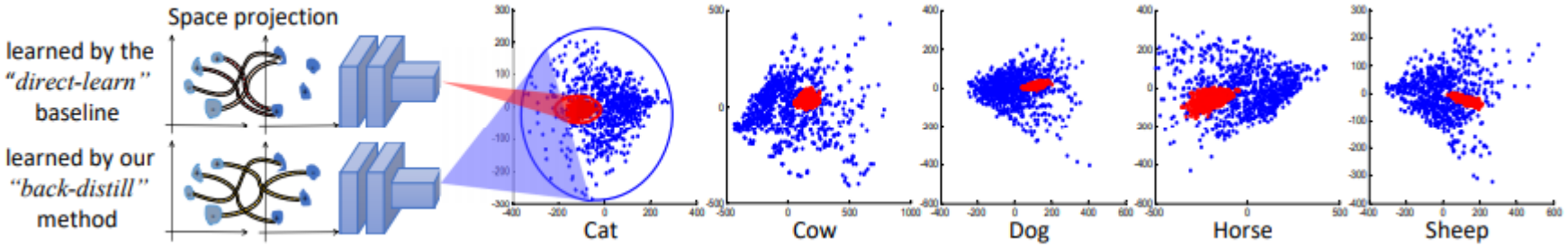
- Beyond sparsity: Tree regularization of deep models for interpretability
  - Regularize RNN model to match decision tree by pathlength estimation
- Ordered Neurons: Integrating Tree Structures into Recurrent Neural Networks (ICLR 2019 Best Paper)
  - Well-designed network architecture
- Dynamic routing between capsules
  - Discuss Later

Network Transplanting



2.3. Application

- Adding adapters to pre-trained CNNs



Insert one conv-layer

# of samples		cat	cow	dog	horse	sheep	**Avg.**
100	direct-learn	12.89	3.09	12.89	10.82	9.28	9.79
	back-distill	**1.55**	**0.52**	**3.61**	**1.55**	**1.03**	**1.65**
50	direct-learn	13.92	15.98	12.37	16.49	15.46	14.84
	back-distill	**1.55**	**0.52**	**3.61**	**1.55**	**1.03**	**1.65**
20	direct-learn	16.49	26.80	28.35	32.47	25.77	25.98
	back-distill	**1.55**	**0.52**	**3.09**	**1.55**	**1.03**	**1.55**
10	direct-learn	39.18	39.18	35.05	41.75	38.66	38.76
	back-distill	**1.55**	**0.52**	**3.61**	**1.55**	**1.03**	**1.65**
0	direct-learn	–	–	–	–	–	–
	back-distill	**1.55**	**0.52**	**4.12**	**1.55**	**1.03**	**1.75**

Insert three conv-layers

# of samples		cat	cow	dog	horse	sheep	**Avg.**
100	direct-learn	9.28	6.70	12.37	11.34	3.61	8.66
	back-distill	**1.03**	**2.58**	**4.12**	**1.55**	**2.58**	**2.37**
50	direct-learn	14.43	13.92	15.46	8.76	7.22	11.96
	back-distill	**3.09**	**3.09**	**4.12**	**2.06**	**4.64**	**3.40**
20	direct-learn	22.16	25.77	32.99	22.68	22.16	25.15
	back-distill	**7.22**	**6.70**	**7.22**	**2.58**	**5.15**	**5.77**
10	direct-learn	36.08	32.99	31.96	34.54	34.02	33.92
	back-distill	**8.25**	**15.46**	**10.31**	**13.92**	**10.31**	**11.65**
0	direct-learn	–	–	–	–	–	–
	back-distill	**50.00**	**50.00**	**50.00**	**49.48**	**50.00**	**49.90**

Table 2. Error rates of classification when we insert n conv-layers with ReLU layers to a CNN as the adapter. $n \in \{1, 3\}$. The last row shows the performance of network transplanting without training samples, *i.e.* without optimizing the task loss $\mathcal{L}(y_S, y^*)$.

2.3. Application

- Transplanting category modules to the classification module

Experiment 2: transplanting to a classification module

# of samples		cat	cow	horse	sheep	**Avg.**	# of samples		cat	cow	horse	sheep	**Avg.**
100	direct-learn	20.10	12.37	18.56	11.86	15.72	20	direct-learn	31.96	37.11	39.69	35.57	36.08
	back-distill	**9.79**	**5.67**	**8.25**	**4.64**	**7.09**		back-distill	**21.13**	**35.57**	**32.47**	**22.68**	**27.96**
50	direct-learn	22.68	19.59	19.07	14.95	19.07	10	direct-learn	41.75	**37.63**	**44.33**	33.51	39.31
	back-distill	**10.82**	**18.04**	**13.92**	**5.15**	**11.98**		back-distill	**34.02**	42.27	44.85	**33.51**	**38.66**

- Transplanting category modules to the segmentation module

Experiment 3: transplanting to a segmentation module

# of samples		cat	cow	horse	sheep	**Avg.**	# of samples		cat	cow	horse	sheep	**Avg.**
100	direct-learn	76.54	74.60	81.00	78.37	77.63	20	direct-learn	71.13	74.82	76.83	77.81	75.15
	distill	74.65	80.18	78.05	80.50	78.35		distill	71.17	74.82	76.05	78.10	75.04
	back-distill	**85.17**	**90.04**	**90.13**	**86.53**	**87.97**		back-distill	**84.03**	**88.37**	**89.22**	**85.01**	**86.66**
50	direct-learn	71.30	74.76	76.83	78.47	75.34	10	direct-learn	70.46	74.74	76.49	78.25	74.99
	distill	68.32	76.50	78.58	80.62	76.01		distill	70.47	74.74	76.83	78.32	75.09
	back-distill	**83.14**	**90.02**	**90.46**	**85.58**	**87.30**		back-distill	**82.32**	**89.49**	**85.97**	**83.50**	**85.32**

2.3. Application

- Transplant to similar or dissimilar categories?

	# of samples		cat	cow	horse	sheep	**Avg.**	# of samples		cat	cow	horse	sheep	**Avg.**
Classification	100	direct-learn	14.43	20.62	17.01	11.86	15.98	20	direct-learn	25.26	24.23	39.18	23.71	28.10
		back-distill	**5.67**	**3.61**	**6.70**	**2.58**	**4.64**		back-distill	**17.01**	**19.59**	**23.71**	**14.95**	**18.82**
	50	direct-learn	21.13	23.71	15.46	10.31	17.65	10	direct-learn	42.27	36.60	40.72	39.18	39.69
		back-distill	**7.22**	**9.28**	**8.76**	**5.67**	**7.73**		back-distill	**42.27**	**32.99**	**28.35**	**30.41**	**33.51**
Segmentation	10	direct-learn	64.97	69.65	80.26	69.87	71.19	20	direct-learn	68.69	81.02	71.88	72.65	73.56
		back-distill	**74.59**	**83.51**	**82.08**	**80.21**	**80.10**		back-distill	**73.34**	**84.78**	**81.40**	**81.04**	**80.14**

Table 4. (top) Error rate of single-category classification when the classification module was learned for both mammals and dissimilar categories. (bottom) Pixel accuracy of object segmentation, the segmentation module was learned for both mammals and dissimilar categories. Other experimental settings were the same as in Experiments 2 and 3. We did not show the result of the dog category, because we needed to compare average performance in this table with results in Table 3.

Thank you