Riemannian Walk for Incremental Learning: Understanding Forgetting and Intransigence

Arslan Chaudhry*, *Puneet K. Dokania**, *Thalaiyasingam Ajanthan**, Philip H. S. Torr
Department of Engineering Science, Torr Vision Group, University of Oxford
*Joint First Authors, **Presented in ECCV 2018**

UNIVERSITY OF OXFORD

Why Incremental Learning?

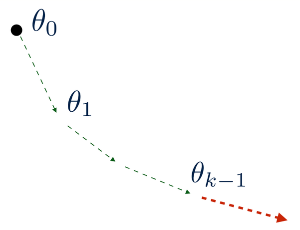
- **Standard Classification**: Given $\mathcal{D} = \{\mathbf{x}_i, \mathbf{y}_i\}_{i=1}^N$, learn mapping $f_\theta : \mathbf{x} \mapsto \mathbf{y}$

$$\min_\theta \mathbb{E}_{\mathbf{x},\mathbf{y}\sim\mathcal{D}} L(f_\theta(\mathbf{x}), \mathbf{y})$$

- **Given a new dataset** $(\bar{\mathcal{D}})$: $\min_\theta \mathbb{E}_{\mathbf{x},\mathbf{y}\sim\mathcal{D}\cup\bar{\mathcal{D}}} L(f_\theta(\mathbf{x}), \mathbf{y})$
  - Not scalable
  - Redundant computations, etc.

Incremental Learning (IL): Problem Definition and Intuitions

Given a sequence of datasets, $\mathcal{D} = \{\mathcal{D}_1, \cdots, \mathcal{D}_k\}$, learn f_θ such that:



$$\min_\theta \mathbb{E}_{\cup_{i=1}^k \mathcal{D}_i} L(f_\theta(\mathbf{x}), \mathbf{y}) \equiv \min_\theta \mathbb{E}_{\mathcal{D}_k} L(f_\theta(\mathbf{x}), \mathbf{y}; \mathbb{K})$$

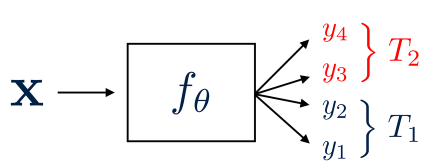
- **What is Knowledge** $(\mathbb{K})$?
  - Input-Output behaviour (knowledge distillation)
  - Parameters of the network
- Preserve Knowledge? Avoid Forgetting
- Update Knowledge? Avoid Intransigence (inability to learn)

Our Contributions

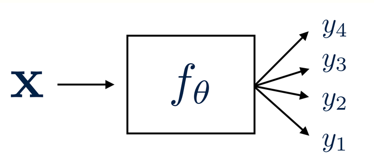
- We observed and defined **intransigence** in IL
- Evaluation settings for IL: **Single-** vs **Multi-head**
- New evaluation metrics: **Forgetting** and **Intransigence**
- Efficient version of EWC [1] which we call **EWC++**
- Generalization of EWC++ and PI [2]: **RWalk**

Incremental Learning Evaluation Setup

- **Multi-head**
  - Task-id is **known** at test time
  - Easier and **unrealistic** setting



- **Single-head**
  - Task-id is **not known**
  - Harder and **realistic** setting



Single-head evaluation should be preferred for IL

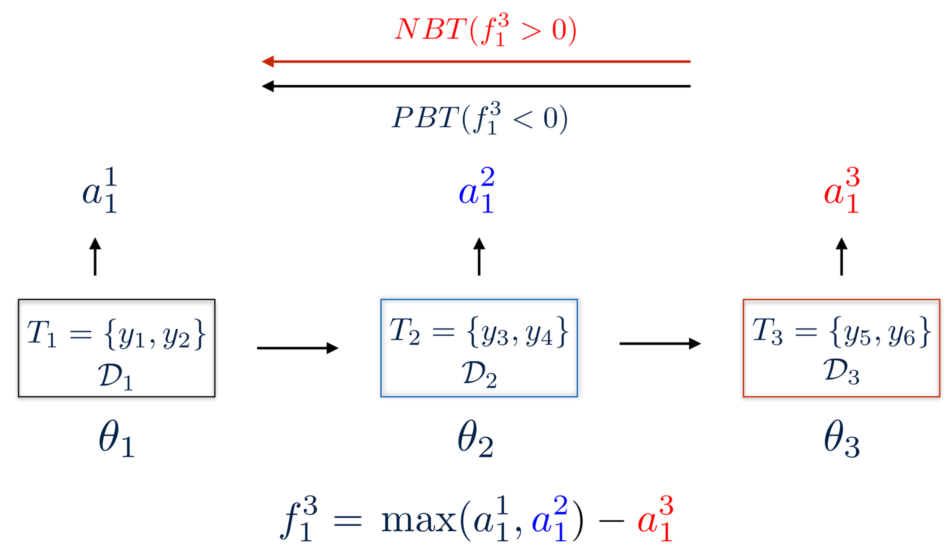
Incremental Learning Evaluation Metrics

Metrics to evaluate IL must consider the path dynamics of training

- **Average Accuracy**: Standard multi-class classification accuracy - does not capture the dynamics of IL
- **Forgetting** and **Intransigence**: capture the IL training path dynamics

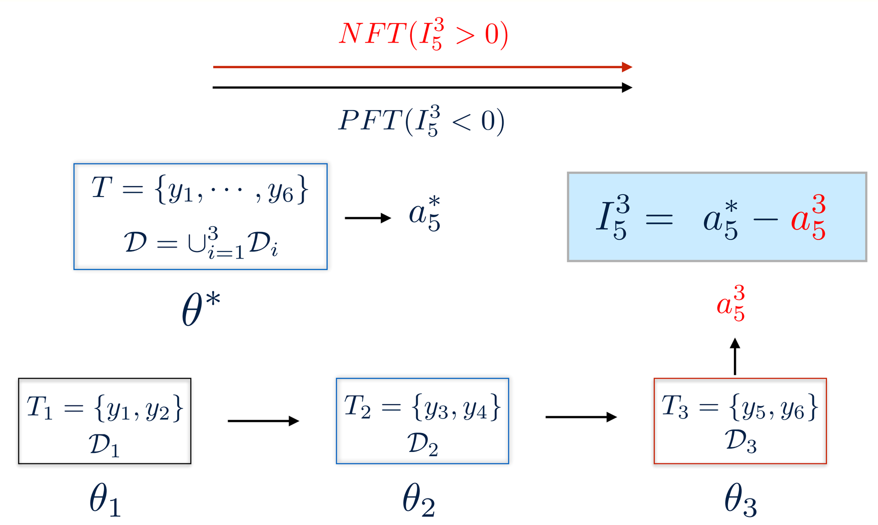
Quantifying Catastrophic Forgetting (Pos/Neg Backward Transfer)

- Influence of learning current task on the *previous* tasks
  - **PBT**: Current task improves performance of previous task
  - **NBT**: Current task reduces performance of previous tasks

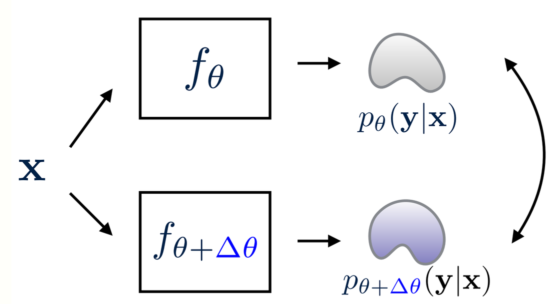


Quantifying Intransigence (Pos/Neg Forward Transfer)

- Intransigence: Inability to learn/update
  - **PFT**: Past tasks improve the performance of the current task
  - **NFT**: Incrementally training till task k reduces its performance



EWC [1]: KL-divergence Perspective



$$D_{KL}(p_\theta \| p_{\theta+\Delta\theta}) \approx \frac{1}{2}\Delta\theta^T F_\theta \Delta\theta \approx \frac{1}{2}\sum_i (\Delta\theta_i)^2 \underbrace{\mathbb{E}_{\mathcal{D}}(g_i^2)}_{F_{\theta_i}}$$

- D_{KL} is equivalent to a distance in a Riemannian manifold induced by F_θ
- F_θ captures the curvature of the KL-divergence surface
- High curvature $\rightarrow$ high sensitivity $\rightarrow$ more important $\rightarrow$ preserve it
- EWC objective: $\mathrm{argmin}_\theta\, L^k(\theta) + \frac{\lambda}{2}\sum_i (\Delta\theta_i)^2 F_{\theta_i}$

EWC++: Efficient Version of EWC [1]

- EWC requires
  - Storing Fisher for each task independently - $\mathcal{O}(kP)$ parameters
  - Additional pass over the dataset to computer Fisher
  - Does not capture influence of parameters along the optimization path
- **EWC++** maintains a single Fisher computed using moving average

$$F_\theta^t = \alpha F_\theta^t + (1-\alpha) F_\theta^{t-1}$$

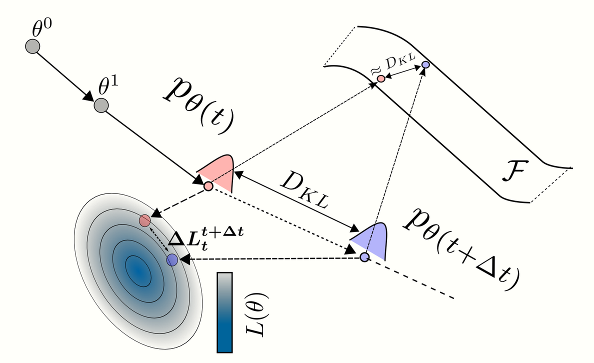
  - Memory Efficient: One estimate of Fisher for all the previous tasks
  - Training Efficient: Fisher computation is the by-product of the training

Optimization Path based Importance

- **Parameter importance**: Change in loss per unit movement in the Riemannian manifold defined by the Fisher Information Matrix

$$s_i = \frac{\Delta L_t^{t+\Delta t}(\theta_i)}{\Delta D_{KL}(\theta_i)}$$

  - Captures the influence of parameters along the optimization path



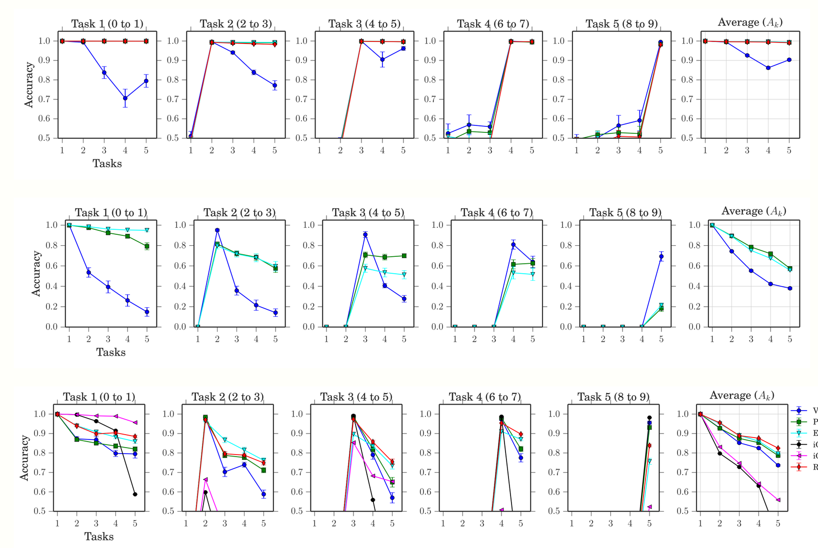
RWalk - Final Objective

$$\tilde{L}^k(\theta) = L^k(\theta) + \lambda \sum_i (F_{\theta_i^{k-1}} + s_{t_0}^{t_{k-1}})(\theta_i - \theta_i^{k-1})^2 \quad (1)$$

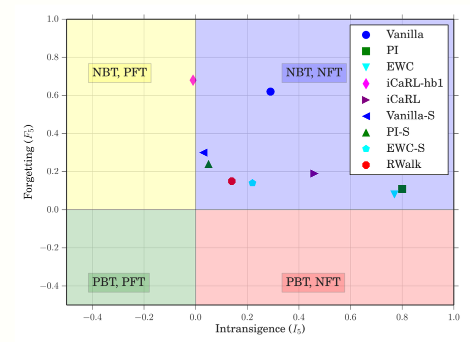
- Eq. 1 is the generalization of EWC++ and PI [2]
  - If no optimization-path based score, then EWC++
  - In Euclidean distance instead of Riemannian, then PI [2]

Results

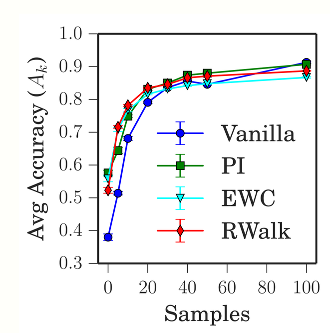
Methods	MNIST				CIFAR			
	Multi-head Evaluation							
	λ	A_5(%)	F_5	I_5	λ	A_{10}(%)	F_{10}	I_{10}
Vanilla	0	90.3	0.12	6.6×10^{-4}	0	44.4	0.36	0.02
EWC	75000	**99.3**	0.001	0.01	3×10^6	72.8	0.001	0.07
PI	0.1	**99.3**	0.002	0.01	10	73.2	0	0.06
RWalk (Ours)	1000	**99.3**	0.003	0.01	1000	**74.2**	0.004	0.04
	Single-head Evaluation							
Vanilla	0	38.0	0.62	0.29	0	10.2	0.36	-0.06
EWC	75000	55.8	0.08	0.77	3×10^6	23.1	0.03	0.17
PI	0.1	57.6	0.11	0.8	10	22.8	0.04	0.2
iCaRL-hb1	-	36.6	0.68	-0.01	-	7.4	0.40	0.06
iCaRL	-	55.8	0.19	0.46	-	9.5	0.11	0.35
Vanilla-S	0	73.7	0.30	0.03	0	12.9	0.64	-0.3
EWC-S	75000	79.7	0.14	0.22	15×10^5	33.6	0.27	-0.05
PI-S	0.1	78.7	0.24	0.05	10	33.6	0.27	-0.03
RWalk (Ours)	1000	**82.5**	0.15	0.14	500	**34.0**	0.28	-0.06



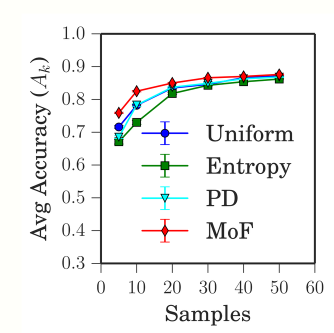
Forgetting and Intransigence Trade-off (MNIST)



F/ I Interplay



Sample efficiency



Different samplings

Acknowledgements

This work was supported by the grants from The Rhodes Trust, EPSRC, SRC and MURI.

References

[1] James Kirkpatrick et al. Overcoming catastrophic forgetting in neural networks. *Proceedings of the National Academy of Sciences of the United States of America (PNAS)*, 2016.
[2] F. Zenke, B. Poole, and S. Ganguli. Continual learning through synaptic intelligence. In *ICML*, 2017.