1 Omnidirectional Transfer for Quasilinear Lifelong Learning

link to paper

1.1 Intro

- Classical ML exhibits **catastrophic forgetting** when trained sequentially. This means that the performance on prior tasks drops when training on new tasks.
- Catastrophic forgetting is not part of biological learning. For example, learning a new language helps performance in native language
- prior work generally is one of two
  1. algo has fixed resources (compresses representatiosn to inorporate new knowledge)(biological equivalent: adulthood -fixed number of cells/synapses)
  2. algo adds/builds resources as new data arrive (biolgocial equivalent: development -adding cells, synapses, etc)
- many SOTA are unable to transfer knowledge forward, and none are able to transfer backward with small sample sizes
- **Omnidirectional learning:** "key innovation is the introduction of representation ensambling which enables omnidirectional transfer vian an omni voter layer, reducing computational time and space from quadratic to quasiliniear"

1.2 Background

1.2.1 Lifelong Learning

generalizes classical ML in a few ways

- instead of one task, there is an environment $\mathcal{T}$ of many tasks
- data arrives sequantially, not batch
- computational complexity constraints on the learning algorithm and hypotheses.

Goal of Lifelong Learning: Given new data and a new task, use all the existing data to achieve lower generalization error on this new task, while also using the new data to obtain a lower generalization error on the previous tasks.

- Previous work relies on continually updating a fixed parametric model or adding resources as new tasks arrive.

1.2.2 Reference Algorithms

Algorithms that build new resources:

- ProgNN
- DF-CNN

Algorithms that leverage fixed resources:

- Elastic Weight Consolidation (EWC)
- Online-EWC (O-EWC)
- Synaptic Intelligence (SI)
- Learning without Forgetting (LwF)
- Total Replay
- Partial Replay

1.3 Evaluation Criteria

- R^t is the risk associated with task t
- $\mathbf{S}_n^t$ is the data from $\mathbf{S}_n$ associated with task t
- $R^t(f(\mathbf{S}_n^t))$ is the risk on task t of the hypothesis learned by f only on task t data
- $R^t(f(\mathbf{S}_n))$ is the risk on task t of the hypothesis learned on all the data
- **Transfer Efficiency** of algorthim f for given task t with sample size n

$$\mathrm{TE}_n^t(f) := \mathbb{E}[R^t(f(\mathbf{S}_n^t))]/\mathbb{E}[R^t(f(\mathbf{S}_n))]$$

  algorithm f has transfer learned for task t with data $\mathbf{S}_n$ iff $\mathrm{TE}_n^t(f) > 1$

- **Forward Transfer Efficiency** of algorithm f for taks t given n samples is

$$\mathrm{FTE}_n^t(f) := \mathbb{E}[R^t(f(\mathbf{S}_n^t))]/\mathbb{E}[R^t(f(\mathbf{S}_n^{<t}))]$$

  an algorithm forward transfers for task t iff $\mathrm{FTE}_n^t > 1$ (the algorithm has used data associated with past tasks to improve performance on task t)

- **Backward Transfer Efficiency** of algorithm f for taks t given n samples is

$$\mathrm{BTE}_n^t(f) := \mathbb{E}[R^t(f(\mathbf{S}_n^{<t}))]/\mathbb{E}[R^t(f(\mathbf{S}_n))]$$

  an algorithm backwards transfers for task t iff $\mathrm{BTE}_n^t > 1$ (the algorithm has used data associated with future tasks to improve performance on previous tasks)

1.4 Omnidirectional Algorithms

1.4.1 Approach

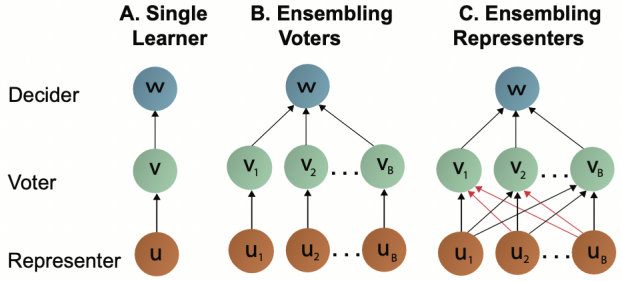

Figure 1: Schemas of composable hypotheses. Ensembling voters is a well-established practice, including random forests and gradient boosted trees. Ensembling representations was previously used in lifelong learning scenarios, but without connections from future tasks to past ones. We introduce such connections, thereby enabling backward transfer.

Lifelong learning relies on hypotheses $h(\cdot) = w \circ v \circ u(\cdot)$

- representor, $u : \mathcal{X} \to \tilde{\mathcal{X}}$ maps an $\mathcal{X}$ valued input into an internal representation space $\tilde{\mathcal{X}}$
- voter, $v : \tilde{\mathcal{X}} \to \Delta_{\mathcal{Y}}$ maps the transformed data into a posterior distribution on the response space $\mathcal{Y}$
- decider, $w : \Delta_{\mathcal{Y}}$ produces a predicted label
- (fig 1B) given B different representors, one can attach a single voter to each representor, yielding B different voters. the decider is then said to ensemble the voters
- (fig1 C) each voter ensembles the representers, and the ensemble of voters feeds into a single decider. This requires an omni-voter layer which ensembles all the existing representations, regardless of the order in which they were learned.

1.4.2 ODIF & ODIN

- As new data from a new taks arrives, ODIF/ODIN first build a new independent representer (using forests or networks), then it builds the voter for this new task, which integrates info across all existing representers, enabling forward transfer. If new data arrive from an old task, it can leverage the new representers to update the voters from the old tasks, thereby enabling backward transfer.
- **ODIF-Omnidirectional Forest**:
  - For each task the transformer u_t of a ODIF is a decision forest
  - the leaf nodes of each decision tree partition the input space $\mathcal{X}$
  - the representation of $x \in \mathcal{X}$ corresponding to a single tree can be a one-hot encoded $L_b$dimensional vector with a 1 in the location of the leaf of tree b that x falls into
  - The transformer u_t is the mapping from $\mathcal{X}$ to a B-sparse vector of lenght $\sum_{b=1}^{B} L_b$
  - Posteriors are learned by poulating the cells of the partitions and taking class votes with out of bag samples. Posteriors output the average normalized class votes across the collection of trees
  - The decider w_t averages the posterior estimates and outputs the argmax to produce a single prediction
- **ODIN-Omnidirectional Deep Network**
  - For each task, the representer u_t maps an element of $\mathcal{X}$ to an element of $\mathbb{R}^d$ where d is the number of neurons in the second to last layer of the network
  - 5 convolutional layers followed by 2 fully connected layers each containing 2,000 nodes with ReLU activation functions and softmax output
  - trained using corss-entropy loss and Adam optimizer to learn the transformer
  - omni voters are learned via KNN
- **ProgNN vs ODIN**
  1. ProgNN builds a new neural network column for each new task, and also builds lateral connections between the new column and previous columns. In contrast, ODIN excludes lateral connections (reducing training time/num parameters)
  2. For inference on task j data, ProgNN only leverages representations learned from tasks up to j whereas ODIN leverahes representations from all J tasks, enabling backward transfer.

1.5 Results & Discussion

Table 1: Capacity, space, and time constraints of the representation learned by various lifelong learning algorithms. We show soft-O notation ($\tilde{\mathcal{O}}(\cdot,\cdot)$ defined in main text) as a function of $n = \sum_t^T n_t$ and T, as well as the common setting where n is proportional to T. Our omnidirectional algorithms are the only algorithms whose representation space grows, but sub-quadratically with n or T, and ODIF is the only algorithm whose time complexity is linear in n for learning the representation.

Parametric	Capacity	Space		Time		Examples
	(n,T)	(n,T)	$(n \propto T)$	(n,T)	$(n \propto T)$	
parametric	1	T	n	nT	n^2	EWC
parametric	1	1	1	n	n	O-EWC, SI, LwF
parametric	1	n	n	nT	n^2	Total Replay
semiparametric	T	T^2	n^2	nT	n^2	ProgNN
semiparametric	T	T	n	n	n	DF-CNN
semiparametric	T	$T+n$	n	n	n	ODIN
nonparametric	n	n	n	n	n	ODIF

- ODIF and ODIN demonstrate possibility of both forward and backward transfer due to leveraging representers learned for other tasks

- To achieve backward transfer, ODIF and ODIN stored old data to vote on the newly learned transformers. Because the representation space scales quasilinearly with sample size, storing the data does not increase the computational complexity of the algorithm, and it remains quasilinear
- representation ensembling approach closely resembles the contructivist view of brain development: the brain goes through progressive elaboration of neural circuits resulting in an augmented cognitive representation while maturing in a certain skill