

1 Omnidirectional Transfer for Quasilinear Lifelong Learning

link to paper

note: need to finish ODIF/ODIN and results

1.1 Intro

- Classical ML exhibit **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 representations to incorporate new knowledge)(biological equivalent: adulthood -fixed number of cells/synapses)
 2. algo adds/builds resources as new data arrive (biological 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 ensembling which enables omnidirectional transfer via an omni voter layer, reducing computational time and space from quadratic to quasilinear"

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 sequentially, 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 algorithm f for given task t with sample size n

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

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

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

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

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

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

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

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

1.4 Omnidirectional Algorithms

1.4.1 Approach

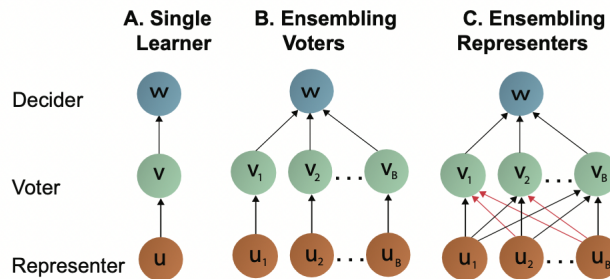


Figure 1: Schemas of composable hypotheses. Ensembling voters is a well-established practice, including random forests and gradient boosted trees. Ensembling representations was previously used in lifelong learning scenarios, but without connections from future tasks to past ones. We introduce such connections,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

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

- representer, $u : \mathcal{X} \rightarrow \tilde{\mathcal{X}}$ maps an \mathcal{X} valued input into an internal representation space $\tilde{\mathcal{X}}$
- voter, $v : \tilde{\mathcal{X}} \rightarrow \Delta_{\mathcal{Y}}$ 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 representers, one can attach a single voter to each representer, 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 task 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
- **ODIF-Omnidirectional 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 leaf of tree b x falls into
 -
- **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 cross-entropy loss and Adam optimizer to learn the transformer
 - omni voters are learned via KNN