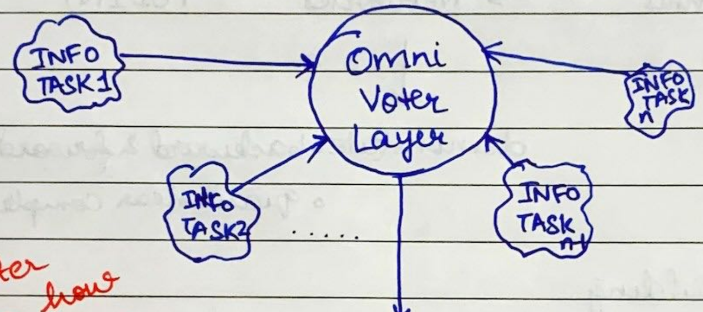
Lifelong learning paper

AIM: Build a bi-directional transfer learning system

DATA
↓
KNOWLEDGE
applicable to past AND future.



- ensembles rep^n learnt independently on all tasks
- decides how to proceed on new data point
- 1 single decision making entity
  ∴ combines all knowledge from all tasks

Q If the omnivoter layer is deciding how to respond, how does it learn "future tasks"?

Catastrophic forgetting: Performance on ~~old~~ tasks falls rapidly as you learn new tasks

Ways to overcome this

Reallocate resources
i.e, forget some things, learn new ones

Add/build resources
i.e, add connec^n ~~based~~ as part of development to make space for new info.

Neither exhibit omnidirectional learning
i.e, most cannot learn forward & none learn backward

* Read ProgNN paper → DeepMind.
* What is quasilinear computational complexity?

Significant step forward:

Progressive NN

new tasks yield additional representational capaci

Omnidirectional learning algorithms → Decision forests (ODIF)

Omnidirectional learning algorithms → Networks (ODIN)

demonstrate
- backward & forward transfer
- quasilinear complexity

Resource building algos.

can leverage prior representations → understand this better.

papergrid

Date: / /

Lifelong learning generalizes classical machine learning by:

i) Environment of $\mathcal{T}$ (possibly ∞) tasks instead of 1

ii) Sequential data instead of batch

iii) Computational complexity constraints

$\therefore$ can't just retrain when new data

GOAL IN LIFELONG LEARNING:

Given new data & a new task,

use all the existing data to achieve lower generalization error on all new & previous tasks

IN CLASSICAL ONLINE SYSTEMS:

Previously experienced tasks may recur,

$\therefore$ maintain / improve performance on those as well

Isn't this true even in lifelong learning?

NOTE: Only looking at task-aware scenarios

i.e, we know all task details for all tasks

i.e, $h : \mathcal{X} \times \mathcal{T} \longrightarrow \mathcal{Y}$

COMPARED TO 9 REFERENCE ALGORITHMS

EVALUATION CRITERIA:

Transfer efficiency = $\dfrac{\text{gen}^n \text{ error of algo trained on data of 1 task } \cancel{t}}{\text{gen}^n \text{ error of algo trained on all data}}$

R^t : risk of task t

S_n^t : data ass^d with task t

S_n : all data

$R^t(f(S_n^t))$ risk on task 't' by hypothesis learnt on ~~all data~~ task t data

$R^t(f(S_n))$ risk on task 't' by hypothesis learnt on all data

$$\text{Transfer efficiency} = \frac{\mathbb{E}(R^t(f(S_n^t)))}{\mathbb{E}(R^t(f(S_n)))}$$

$TE_n^t(f)$

Algorithm 'f' has transfer learned for task 't' with data 'S_n' if & only if $TE_n^t(f) > 1$

Respecting the 'streaming' nature of tasks:

task info. known for all, but data appears sequentially

Forward transfer efficiency = $\dfrac{\text{Risk of algo. with access to data from 't'}}{\text{Risk of algo. with access upto task 't'.}}$
($FTE_n^t(f)$)

& including last obs^n of.

$$= \frac{\mathbb{E}(R^t(f(S_n^t)))}{\mathbb{E}(R^t(f(S_n^{<t})))}$$

An algo. has used data from prv. tasks to improve perf. on t if FTE>1

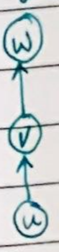
Backward transfer efficiency = $\dfrac{\mathbb{E}(R^t(f(S_n^{<t})))}{\mathbb{E}(R^t(f(S_n^t)))} = \dfrac{\text{Risk with access upto 't'}}{\text{Risk with access to 't'}}$
($BTE_n^t(f)$)

An algo. has used data from future tasks to improve perf. on t if BTE>1 papergrid

Date: / /

Omnidirectional Algorithms

for single learner



$$h(\cdot) = w \circ v \circ u(\cdot)$$

hypotheses for lifelong learning, can be decomposed into 3 constituent parts

$\boxed{\text{decider}}$ — $\Delta y \to Y$ — produces a predicted label

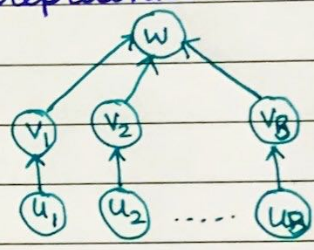
$\boxed{\text{voter}}$ — $\tilde{X} \to \Delta y$ — maps transformed data to a posterior distribution i.e, a score on response space Y.

$\boxed{\text{representer}}$ — $u: X \to \tilde{X}$ — maps i/p X to internal rep^n space $\tilde{X}$

Step 1

'B' diff representers → one voter per representer → B voters

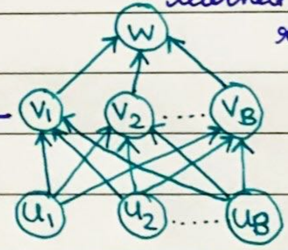
Decider ensembles voters



Step 2:

Each voter ensembles representers → helps if each representer has learned complementary representations

$\underbrace{\text{"Omni-voter layer"}}$ → ensembles all the existing representations, irrespective of the order they were learnt in



Then, ensemble of voters feeds into the decider → ~~helps of~~

* Omnidirectional Forest (ODIF) : Decision forest based instance of ensembling representations

* Omnidirectional Network (ODIN) : Deep network based instance of ensembling representations