No: 76

Continual Learning with Self-Organizing Maps

Pouya Bashivan[1], Martin Schrimpf[1], Robert Ajemian[1], Irina Rish[2,3], Matthew Riemer[2,3], Yuhai Tu[2,3]

[1] McGovern Institute for Brain Research, Massachusetts Institute of Technology

[2] MIT-IBM Watson AI Lab [3] IBM Research

ABSTRACT

- When exposed to non-stationary learning environments, current neural networks tend to forget what they had previously learned, a phenomena known as **catastrophic forgetting**.
- Most previous approaches to this problem rely on **memory replay buffers** which store samples from previously learned tasks, and use them to regularize the learning on new ones.
- This approach suffers from the important disadvantage of **not scaling well to real-life problems** in which the memory requirements become enormous.
- We propose a **memoryless** method that combines standard supervised neural networks with self-organizing maps to solve the continual learning problem.

METHOD

- Self-Organizing Maps (SOMs) [1] create a low-dimensional representation of the input w/o supervision.
- Competitive learning allows the SOM to **only adapt parts of its parameters** to each input pattern that creates an opportunity for using it in continual learning setting.
- A SOM layer is trained unsupervised in parallel to the MLP layer and functions as **multiplicative mask** on MLP outputs.

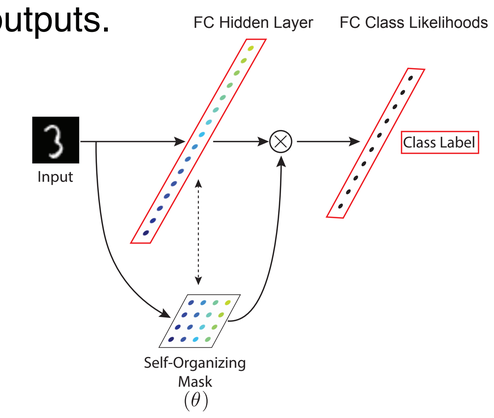
Algorithm 2.1 SOM update

Given: input x and SOM weights θ at step t, nodes coordinate matrix L, batch size N_{bs}, number of nodes N_n, and hyperparameters $\alpha, \sigma, \epsilon, \tau$.

$\bar{x} = \frac{1}{N_{bs}} \sum_{i \in batch}^{N_{bs}} x_i$ ▷ average over batch
▷ Find best matching unit (BMU) for $\bar{x}$
$i^* = argmin_i(\|\bar{x} - \theta_i\|)$
for $i = 1, 2, ..., N_n$ **do** ▷ compute
  $D_i = \|L_i - L_{i^*}\|$ ▷ distances to BMU
end
$\phi = e^{D/\sigma^2}$ ▷ neighborhood mask
$\Gamma = e^{-\|x-\theta\|/\epsilon}$ ▷ output mask
$\alpha \leftarrow \alpha e^{-\tau t/N_{steps}}$ ▷ Adjust σ and α
$\sigma \leftarrow \sigma e^{-\tau t/N_{steps}}$
$\theta_i \leftarrow \theta_i + \alpha\phi(x - \theta_i)$ ▷ Update weights

Algorithm 2.2 Training procedure

Given: Training datasets (X_t, Y_t) for each task t out of N_T tasks, and data subsets (X_t^{pr}, Y_t^{pr}) available for SOM pretraining.
Initialize: SOM and MLP weights randomly.
▷ Pretrain sequentially w/ unlabeled data
for $t = 1, 2, ..., N_T$ **do**
  for batch in X_t^{pr} **do**
    Update SOM weights using Alg. 2.1
  end
end
▷ Train sequentially using labeled data
for $t = 1, 2, ..., N_T$ **do**
  for batch in (X_t, Y_t) **do**
    Update SOM weights using Alg. 2.1
    Update MLP weights using SGD
  end
end

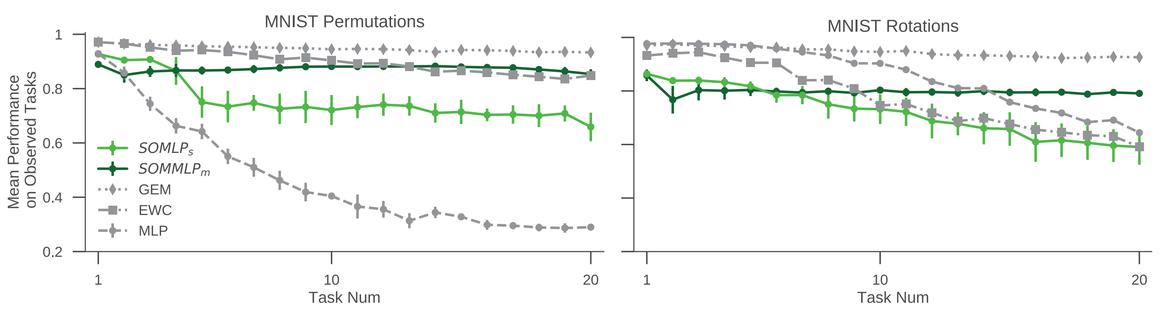


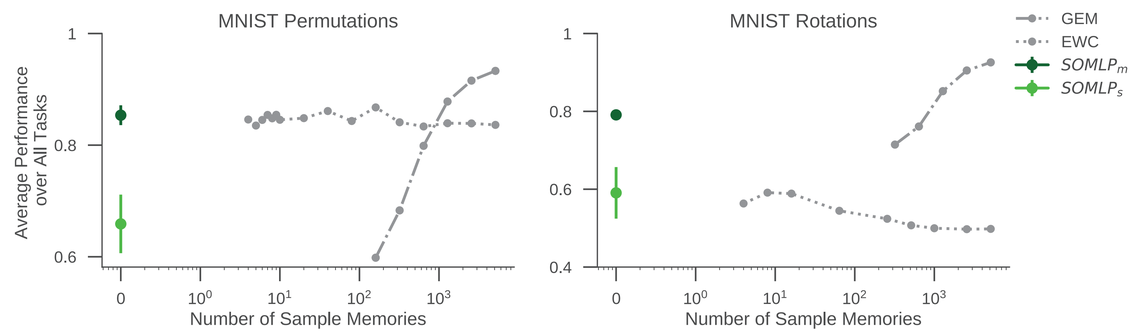
RESULTS

- We equalized the number of trainable parameters in all networks (2.5 M).
- SOMLP with random initialization does not perform well on either tasks but allowing **pretraining on limited unlabeled data** (SOMLP$_m$) leads to outperfoming GEM [2] and EWC [3] in low-memory regime on both tasks.

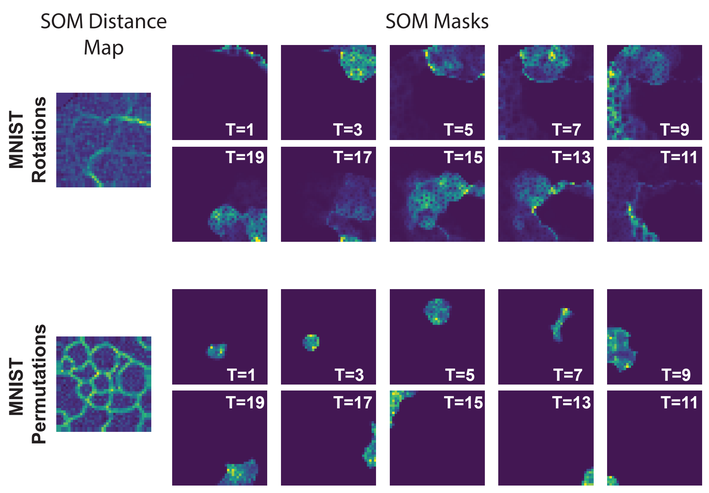
Network	Memory Size	#Parameters (M)	Performance (%) (MNIST-Permutations)	Performance (%) (MNIST-Rotations)
MLP	0	2.5	29 ± 0.6	64.4 ± 0.4
EWC-Low Mem	8	2.5	84.8	59.1
EWC-High Mem	5120	2.5	83.6	49.8
GEM-Low Mem	320	2.5	68.3	71.5
GEM-High Mem	5120	2.5	93.3	92.6
SOMLP$_s$	0	2.5	65.9 ± 5.2	59.1 ± 6.6
SOMLP$_m$	0	2.5	85.4 ± 1.8	79.1 ± 0.8

- SOMLP$_m$ is robust to forgetting on both benchmark tasks, with almost no degradation in average performance throughout the procedure.
- SOMLP starts from a lower performance level, possibly because of smaller MLP layer (half-sized).
- GEM performance is dependent on the amount of replay memory, and quickly degrades with decreasing memroy buffer size.
- EWC is less dependent on the number of memroy slots but does not perform well on MNIST-rotations with our experiment setup.





- On MNIST-rotations task, SOMLP shares its resources between similar tasks. Visual inspection of per-task masks, suggests that the amount of overlapping between closer tasks is higher.
- On MNIST-permutations, SOMLP learns separate masks for each task. This leads to tiling the resources into separate areas.



IMPLICATIONS

Upsides

- Memoryless method that performs better than GEM and EWC method in low-memory setting.
- Robust to forgetting when pretrained on some data.
- Unlike GEM that uses the task-information to retain samples, our method does not use task-information in any way.

Downsides

- Needs pretraining on unlabled data to perform well.
- Needs a large network to solve many tasks in continual learning setting.
- Performs lower than state-of-the-art methods with large memory buffers.

REFERENCES

1. Kohonen, T. (1990). The Self-Organizing Map. Proceedings of the IEEE. https://doi.org/10.1109/5.58325
2. Lopez-Paz, D., & Ranzato, M. (2017). Gradient Episodic Memory for Continual Learning. Neural Information Processing Systems. Retrieved from http://arxiv.org/abs/1706.08840
3. Kirkpatrick, J., Razvan, P., Rabinowitza, N., Venessa, J., Desjardinsa, G., Rusua, A. A., … Hadsella, R. (2017). Overcoming catastrophic forgetting in neural networks. PNAS. https://doi.org/10.1073/pnas.1611835114

FUTURE WORK

- Test it on unbalanced datasets
- Study the effect of network size on performance
- Extension to multi-layer SOMLP

ACKNOWLEDGEMENT

This research was supported by Intelligence Advanced Research Projects Agency (IARPA) and the MIT-IBM Watson AI Lab.