Catastrophic Forgetting: An Extension of Current Approaches

Dhrupad Bharadwaj, Evaristus Ezekwem, Angela V. Teng
Advised by: Dr. Julia Kempe, Artem Vysogorets

DSGA1006: Capstone
Fall 2020

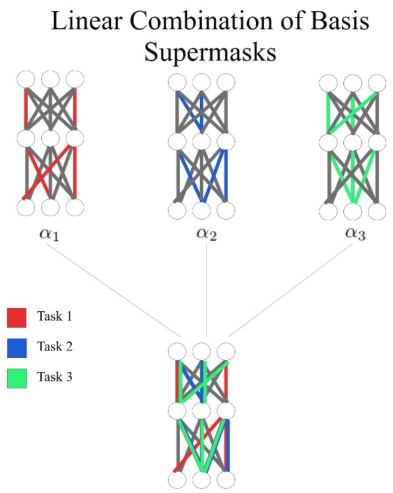
NYU Center for Data Science

Abstract

Catastrophic forgetting is the phenomenon whereby the performance of learned tasks degreades as new, unseen tasks are learned, specifically in Neural Networks. We extend Wortsman et al.'s (2020) work on continual learning, *Supermasks in Superposition*, by adapting their masking technique to learn new tasks while using significantly fewer additional parameters.

Introduction and Background

Catastrophic forgetting is an active area of research in continual learning. In order to achieve artificial general intelligence (AGI) it is crucial that learning models are able to learn and remember a wide variety of tasks. Deep learning models have a tendency to forget older tasks once new ones are learnt.



Research done by Wortsman et al. propose the use of Supermasks in Superposition (SupSup), which is "capable of sequentially learning thousands of tasks without catastrophic forgetting." [Wortsman et al., 2020]

Our research extends this approach of using randomly initialized and fixed base networks for each task by optimizing over a linear combinations of supermasks, referred to as the basis masks, to learn new tasks.

In general, we find that:

- a mask applied to an arbitrary task t performs no better than the maximum entropy distribution over the output space
- a combination of masks are able to effectively learn new tasks

- while performance is slightly inferior to SupSup, our approach only requires *O(nd)* parameters for each new task, where *d* is the depth of the network and *n* is the number of masks used.

This is a considerable improvement in parameter efficiency compared to SupSup, where each additional task requires storing *O(w)* parameters, where *w* is the number of trainable weights in the network.

References

[1] Ramanujan, Vivek, Mitchell Wortsman, Aniruddha Kembhavi, Ali Farhadi, and Mohammad Rastegari.2019. "What's Hidden in a Randomly Weighted Neural Network?" Proceedings of the IEEE Computer SocietyConference on Computer Vision and Pattern Recognition, November, 11890–99. http://arxiv.org/abs/1911.13299

Methods and Algorithm

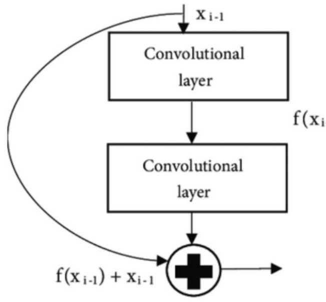
Our problem to **find the optimal α_i* for a given task $i \in T'$** can be formulated as follows:

$$\alpha_i^* = \arg\min_{\alpha_i} \; \mathcal{L}\left(y, f\left(x, \left(\frac{1}{|B|}\sum_{t\in T^B}\sum_{d\in D}\alpha_i^{td}\Delta(M_t,d)\right)\odot W\right)\right)$$

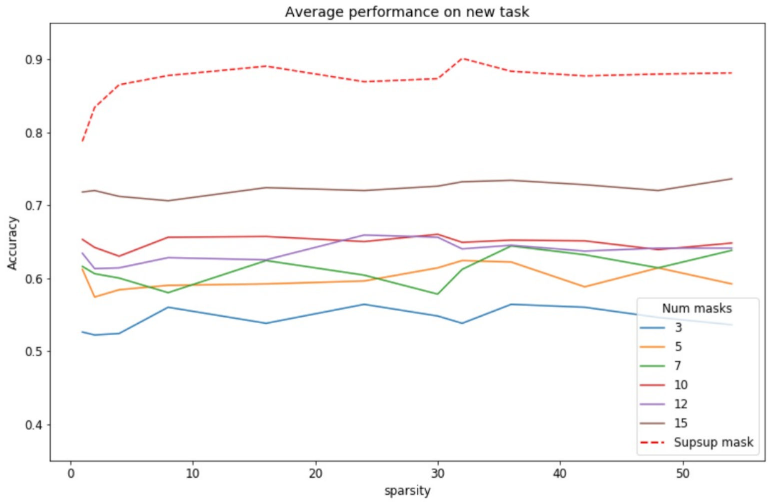
Where B = basis tasks, T = all set of tasks, Mi = supermask for task i, W = weight matrix, d = layer of the network, D = depth of the network

Extending the SupSup model, we take a set of masks $B \subset M^*$, our set of basis masks, and define the set of tasks for which we have a trained mask set, $T^B := \{t \mid M_t^* \in B, \forall\, t \in T\}$ and $T' := T \setminus T^B$.

We define $\alpha_i \in \mathrm{R}^{d\times|B|}$ where we have one parameter per $M_t^* \in B$ per layer of the network. Let α_i^{td} be the parameter for task $i \in T'$ on $\Delta(M_t^*, d)$, where $\Delta(M,d)$ is the bit-mask from M in corresponding layer *d* of the network. D is the depth of the network.



Results



- Generally, more masks are better
- Weak inverse relationship with mask sparsity
- Faster convergence compared to SupSup
-10x reduction in parameter overhead per task: 96875 vs 315 on ResNet-18 alone
- Savings more dramatic in wide architectures

Benchmark: Supsup model evaluated on random 5-way CIFAR100 task
Network: ResNet-18
Data: 5-way CIFAR100 task
Details:
- Learning Rate: 0.1
- Adam optimizer
- Cross Entropy Loss
- PyTorch and Torchvision
- 250 epochs at batch size of 64
- NVIDIA K80 GPUs
- L2 Regularization on Alphas

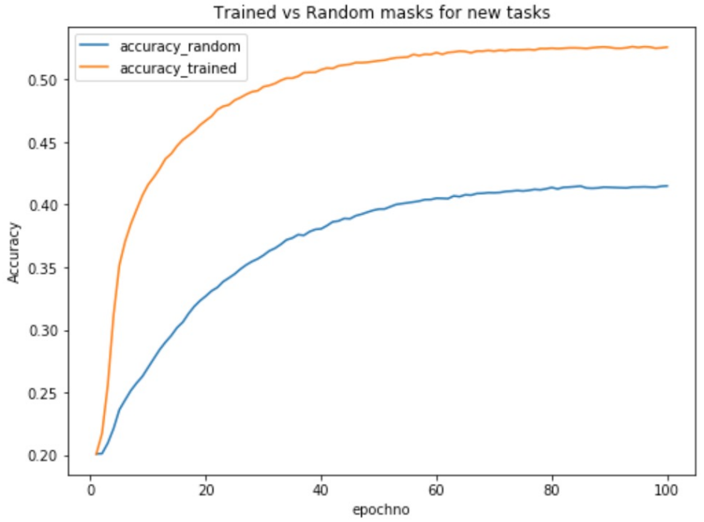
Evaluated using two kinds of seed models:
- SupSup model with random masks
- SupSup model with trained masks

Note: The full code along with our paper and full set of references may be viewed here: https://github.com/DSGA1006-Capstone-Team/supsup

Model	#params	Conv Layers	#values/mask (Supsup)	#values/new task (15 masks)
ResNet-18	6.2M	21	96875	315
Wide ResNet-18	11.7M	21	182813	315
Wide ResNet-34	21.8M	37	340625	555
ResNet-50	25.6M	53	400000	795
ResNet-101	44.5M	104	695313	1560
ResNet-152	60.2M	155	940625	2325
WRN-50-2-bottleneck	68.9M	53	1076563	795
pre-ResNet-200	64.7M	203	1010938	3045

An intuitive way to think about the algorithm is through neural network pathways. Inherently, supermasks either activate our deactivate neural network pathways to determine which weights are applied to a given task and which connections from the dense network are preserved. Each SupSup mask corresponds to one such set of pathways over the fixed-weight backbone network. Our approach optimizes the weighted sum of these pathways, and stores a new weight vector for each additional task learned.

- Elements of transfer learning are observed: (1) Using random masks is significantly less effective (2) More masks narrows the gap
- SupSup performance correlated with basis mask performance
- Sparsity effect: (1) Less overlap between masks improves predictiveness of linear combinations (2) Less "bad" weights/ connections bundled in



Conclusions and Future Work

Outcomes

- New algorithm is able to effectively learn new tasks
- Theoretically unbounded tasks possible to be learned
- Performance on certain tasks competitive with stand-alone smaller architectures
- Parameter savings over 100x on SupSup

Future Research

- Wider networks would perform better based on the (1-q)S rule *(Ramanujan et al. (2020)*, experiment with WRN-34
- Explore using a trained backbone network + randomly initialized seed masks
- Try different task partitioning techniques: Masks on PCA / NMF components