Continual Learning With Mixture of Experts using subnetwork selection and masking

Harshal Kulkarni[*]
Department of Electrical and Computer Engineering
New York University
Brooklyn, NY 11201
hsk8171@nyu.edu

Abstract

Existing research in the task-agnostic continual learning setting use loss function value as an indicator for the onset of a new task and hence creating a new instance of an expert for the incoming new task. Although this prevents catastrophic forgetting, these methods are highly inefficient and prevent knowledge transfer. The goal of the proposed method is to create diverse models over time, where each model focuses on training tasks with similar parameter spaces (benefitting from knowledge transfer and catastrophic remembering) along with alleviating catastrophic forgetting. The gradients of the loss with respect to the parameters after convergence are used to identify the useful subnetwork for the task at hand, and freeze them for training the new task on the remaining subnetwork, enabling knowledge transfer and alleviating catastrophic forgetting, creating an efficient system as a whole.

1 Introduction

Catastrophic forgetting (CF) and knowledge transfer (KT) are two key challenges of continual learning (CL), which learns a sequence of tasks incrementally. CF refers to the phenomenon in which a model loses some of its performance on previous tasks once it learns a new task. KT means that tasks can help each other learn by sharing knowledge. Catastrophic Forgetting is largely considered a direct consequence of the overlap of distributed representations in the network. Most prior work deals with CF by either completely removing the representational overlap (French, 1991; Kirkpatrick et al., 2017) or, more frequently, by replaying data from previous tasks. Data replay methods can deal with CF but, in turn, lead to a reduced capacity of the network to discriminate between old and new inputs (Sharkey & Sharkey, 1995b). This is called Catastrophic Remembering (CR) and has been shown to be a significant limitation of replay methods (Robins, 1993; Sharkey & Sharkey, 1995b; Kaushik et al., 2021).

To better demonstrate understanding of CR and why CF alleviation aggravates, we take probabilistic view of the problem: Let θ_i be the initial parameters for training model on D_i dataset. We can compute the conditional probability of the first task $P(\theta_1|D_1)$ from the prior probability of the parameters $P(\theta_1)$ and the probability of the data $P(D_1|\theta_1)$ by using Bayes' rule. Hence, for the first task,

$$\log P(\theta_1|D_1) = \log P(D_1|\theta_1) + \log P(\theta_1) - \log P(D_1) \quad (1)$$

[*] Special thanks are due to Professor Anna Choromanska for her regular discussions and insightful ideas. Github code repository: https://github.com/HARSHALK2598/Continual-Learning-with-Mixture-of-Experts

If we were to now train the same network for a second task, the posterior from (1) now becomes a prior for the new posterior. If no regularization or method is included to preserve the prior information, we'd optimize for the second task:

$$\log P(\theta_{1:2}|D_{1:2}) = \log P(D_2|\theta_2) + \log P(\theta_1|D_1) - \log P(D_2) \quad (2)$$

$$\log P(\theta_{1:2}|D_{1:2}) = \log P(D_2|\theta_2) + \log P(D_1|\theta_1) + \log P(\theta_1) - \log P(D_1) - \log P(D_2) \quad (3)$$

For nth task, we have

$$\log P(\theta_{1:n}|D_{1:n}) = \log P(D_n|\theta_n) + \sum_{i=1}^{n-1} \log P(D_i|\theta_i) + \log P(\theta_1) - \sum_{i=1}^{n} \log P(D_i)$$

When from right hand side $\log P(D_n|\theta_n) << \sum_{i=1}^{n-1} \log P(D_i|\theta_i)$, i.e as model is trained on multiple tasks, eventually it looses its discriminative ability. As new task is introduced there is false sense of familiarity of task which prevents model to learn the new feature space.

This work investigates these problems in the less popular CL paradigm, task-agnostic incremental learning (TAIL). In TAIL, each task consists of several classes of objects to be learned. Once a task is learned, its data is discarded and will not be available for later use. During testing, the task ID is not provided for each test sample; therefore, corresponding identification of the task ID is an additional overhead.

2 Related Work

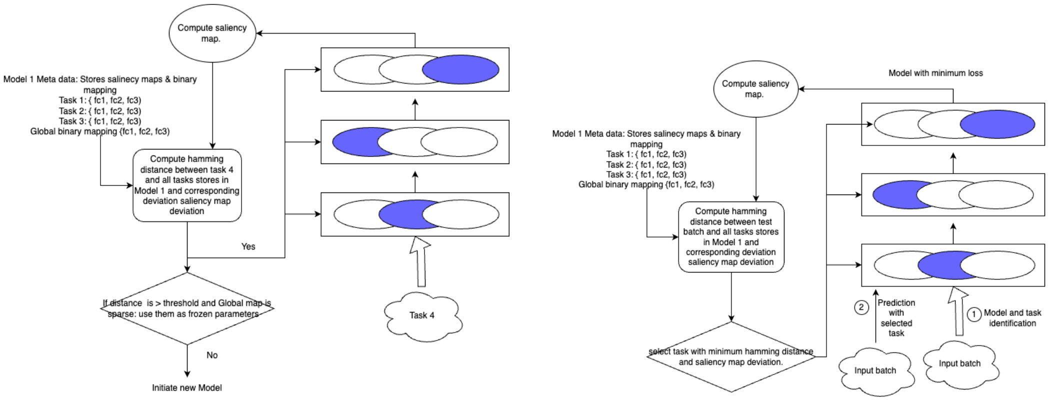
Several effective Task incremental Learning (TIL) approaches minimize CF by parameter isolation, masking sub-networks for each task within a shared network. HAT (Serra et al., 2018) and SupSup (Wortsman et al., 2020) are representative systems. HAT applies hard masks on task-crucial neurons, obstructing gradient flow through masked neurons during new task learning. Only unmasked neurons and parameters are trainable. As tasks increase, available neurons decrease, leading to performance deterioration and capacity issues if a neuron is masked. CAT (Ke et al., 2020) improves HAT's KT by detecting task similarities and removing masks of similar previous tasks for new task training. However, risks include severe CF if dissimilar tasks are wrongly deemed similar, and limited KT if similar tasks are missed. SupSup uses a randomly initialized backbone, identifying sub-networks (masks) for each task independently. No CF or capacity issues, but by design, no KT since masks are independent.

Konishi et al. (2023) recently proposed SPG method. Instead of learning hard/binary masks on neurons for each task and blocking these neurons in training a new task and in testing like HAT, SPG computes an importance score for each network parameter (not neuron) to old tasks using gradients. The reason that gradients can be used as an importance is because gradients directly tell how a change to a specific parameter will affect the output classification and may cause CF.

We are the first ones to propose parameter isolation for the TAIL setting. We utilize the gradient scores after convergence to identify the selection of an optimal subnetwork for a task in a selected model, as well as to identify the onset of a new task. Features are extracted at the parameter level, not at the neuron level. Because there is more noise when sampling neurons for the subnetwork, hence more parameters are sampled to maintain task accuracy. The proposed method uses parameter-based masking similar to SPG, but with a task-agnostic Mixture of Experts (MoE) setting. Based on the similarity with the parameter space of trained tasks and constructive overlapping regions, we identify model to for training a new task and similar old tasks subnetwork, previously trained on same model as baseline for knowledge transfer otherwise intiate a new model.

3 Proposed Method

Refer Figure 1, for complete architecture and flow of dataset through selected network during training and test.


Compute saliency map.
Model 1 Meta data: Stores salinecy maps & binary mapping
Task 1: { fc1, fc2, fc3)
Task 2: { fc1, fc2, fc3)
Task 3: { fc1, fc2, fc3)
Global binary mapping {fc1, fc2, fc3)
Compute hamming distance between task 4 and all tasks stores in Model 1 and corresponding deviation saliency map deviation
Yes
If distance is > threshold and Global map is sparse: use them as frozen parameters
No
Initiate new Model
Task 4
Model with minimum loss
Compute saliency map.
Model 1 Meta data: Stores salinecy maps & binary mapping
Task 1: { fc1, fc2, fc3)
Task 2: { fc1, fc2, fc3)
Task 3: { fc1, fc2, fc3)
Global binary mapping {fc1, fc2, fc3)
Compute hamming distance between test batch and all tasks stores in Model 1 and corresponding deviation saliency map deviation
select task with minimum hamming distance and saliency map deviation.
2
Prediction with selected task
1
Model and task identification
Input batch
Input batch

(a) Flow during onset of a new task.

(b) Flow during testing an input batch

Figure 1: Left side: Model is already trained on 3 tasks and corresponding subnetwork binary mapping and gradient scores are stored as meta data. When new task dataset comes in online setting, onset of new task is detected and previously trained tasks with similar parameter space are selected based on overlap (hamming distance between binary mappings) and corresponding gradient score deviation. Right Side: is the flow input batch during test/prediction procedure.

3.1 Importance of gradients after convergence.

Gradients of loss with respect to the parameters highlight the network parameters responsible in minimizing loss in optimization landscape. Network parameters with high value of gradient value needs to be preserved for retaining the knowledge of trained task and hence mitigate catastrophic forgetting.

3.2 Subnetwork selection for a given task

As shown in Figure 1.a, for a given task, after convergence, we identify the subnetwork based on the gradients of loss over the last batch of training dataset. We mask these subnetwork for training new tasks to alleviate catastrophic forgetting. We normalize the gradients over each layer and select parameters with values greater than threshold. Threshold is hyperparameter, adjusted to preserve the accuracy for the task over selected subnetwork. For each trained task, we maintain the subnetwork gradient scores after convergence, and corresponding binary mapping. This is used later to match the new tasks overlap (using hamming distance between old tasks binary mapping and new task binary mapping) and check the overlap is constructive based on difference in gradient scores. Naturally if overlap gradient divergence is huge, its an indicator of disruptive overlap. In case of constructive overlap we utilise old tasks for training new task, enabling knowledge transfer.

3.3 Onset of new task detection and training

With the chosen sub network for the old task, new task shows significant gradient deviation when compared to the difference between images belonging the old task. This is used as a measure to identify the onset of a new task. When a new task is detected, we identify the parameters with gradient scores more than threshold value, and corresponding binary mapping. We use these to identify the most similar tasks previously trained on same model as explained in section 3.2. These old tasks sub networks are utilised for knowledge transfer in training of new task. Backward gradient flow for old tasks sub network is restricted. This helps in knowledge transfer along with alleviating catastrophic forgetting for pre-trained tasks.

3.4 Initiate new Model

Each model stores list of tasks trained on that model, mapped to it corresponding subnetwork binary mapping and gradient scores. If with new task overlap with previously trained tasks less than threshold or the overlap is destructive (significant deviation of gradients over the overlapped region) we create a new instance of model for training the new task.

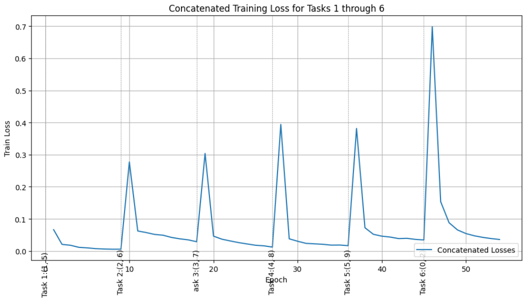
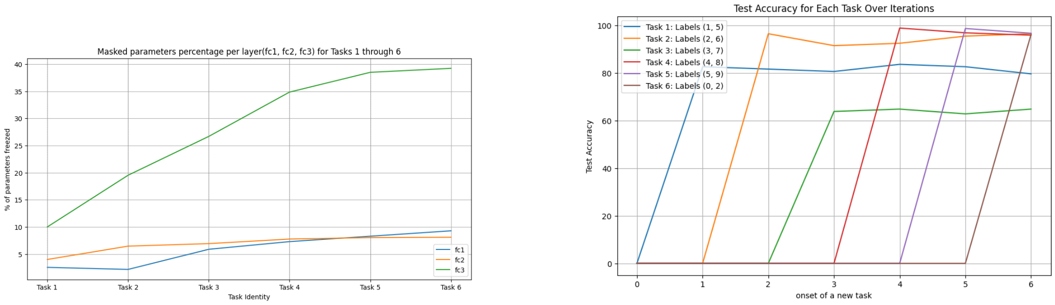


Figure 2: Loss function value as new tasks are trained on model when part of the network is frozen for alleviating catastrophic forgetting on previously trained tasks.



(a) Total network frozen as new tasks are trained on exisiting models.

(b) Test Accuracies for each task trained on same model over a period of time.

Figure 3: Left side shows model sparsity/percentage of network frozen as new tasks are introduced, right side, shows the tasks accuracies as new tasks are trained on same model after optimally freezing sub networks for older tasks.

3.5 Testing phase

As shown in Figure 1.b, We create two instances for the same batch during test phase. First batch is used for selecting the best model and best task among all tasks trained on that model. Test batch is passed through all models, and model with least loss is selected. In the selected model, we identify the task subnetwork based on the gradient score binary mapping overlap and overlap gradient score deviation. Once the model and corresponding subnetwork is selected, we use the second instance of batch for best prediction.

4 Experiment and Results

We perform experiments with split mnist dataset. We divide minst dataset into 6 binary classification tasks task 1: Labels (1, 5); task 2: Labels (2, 6); task 3: Labels (3, 7), task 4: Labels (4, 8), task 5: Labels (5, 9), task 6: Labels (0, 2). Each model is a dense neural network with 3 layers. First layers has (500, 784) parametes, second layer has (500, 500) parameters and last layer has (10, 500) parameters. Each task is fed to the model in online fashion, sequentially. For training model on each task till convergence we feed the model each task for multiple epochs (assumption is each task has sufficient data for model to reach optimmum convergence before introduction of a new task).

We train the model on task 1, identify optimal subnetwork after convergence. Then we freeze this subnetwork and introduce next task. Refer Figure 2, which shows introduction of new task and corresponding decrement in loss function values. Although we see significant change in loss function, it does not convey any information about how much parameter space is shared within 2 tasks.

We continue above training process, and analyse the percentage of network frozen as new tasks are trained on same model. As shown in figure 3.a, we see first 2 layers use less than 10% of the network. This implies 2 things.

Gradient score change	Task 1	Task 2	Task 3
Layer 1	0.385	-4856.520	-6907.973
Layer 2	-0.034	-702.947	-973.901

Table 1: For an input test image from task 1, table shows difference between gradient scores for input image with each task gradient scores. There is minor deviation for task 1 when compared to task 2 and task 3.

- For each task we need to freeze only a small percentage of network for retaining the test accuracy more than 75%, leaving room for more tasks to be trained on the network model.
- As new tasks are trained, except the last layer, percentage of the network frozen increases by less than 20%. There is huge contructive overlap between parameter spaces of new tasks introduces and hence new tasks are benefiting from knowledge transfer from old tasks.

We analyse the test accuracy for all the trained tasks on a particular model. As shown in Figure 3.b, for task 1, training starts at 0, and we introduce new task at 1. Test accuracy for task 1 at point 1, is more than 80%. This accuracy is from only optimal subnetwork selected. As this subnetwork is frozen during the training of the next tasks, there is very less deviation in test accuracy at later stage. Same happens with other tasks introduces at later period.

Finally we anaylse how to identify correct task given input test image. As show in table 1, model is trained on 3 tasks. As we get input image for task 1. We compare the gradient score deviation of test image with task 1, task 2 and task 3 subnetwork gradient scores. As we see there is huge deviation for task 2 and task 3 subnetwork gradient scores and hence task 1 subnetwork is correctly selected for making prediction.

5 Conclusion

In conclusion, we propose a new parameter isolation method on task-agnostic based continual learning setting. As the algorithm proceeds, over a period of time, as we choose the models for training tasks based on the parameter space for the old tasks trained on the same model, eventually we are creating most diverse models, where each model is responsible for training tasks sharing maximum parameter space fully utilising knowledge transfer at the same time effectively alleviating catastrophic forgetting. This is a first method to propose analysis on similarity based diversification on Mixture of Experts (normally it's done based on loss on input data). unlike any work done in task agnostic continual learning setting, this method does not require storing any form of input data (strictly adhering to continual learning settings) and can be extended to CNNs.

References

[1] Konishi, T., Kurokawa, M., Ono, C., Ke, Z., Kim, G., Liu, B.: Parameter-level soft-masking for continual learning. In: Krause, A., Brunskill, E., Cho, K., Engelhardt, B., Sabato, S., Scarlett, J. (eds.) *International Conference on Machine Learning, ICML 2023, 23-29 July 2023, Honolulu, Hawaii, USA. Proceedings of Machine Learning Research*, vol. 202, pp. 17492–17505.

[2] James Kirkpatrick, Razvan Pascanu, Neil C. Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A. Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, Demis Hassabis, Claudia Clopath, Dharshan Kumaran, and Raia Hadsell. 2016. Overcoming catastrophic forgetting in neural networks. *CoRR.*

[3] Serra, J., Suris, D., Miron, M., and Karatzoglou, A. Overcoming Catastrophic Forgetting with Hard Attention to the Task. *In Proc. of ICML*, 2018.

[4] Wortsman, M., Ramanujan, V., Liu, R., Kembhavi, A., Rastegari, M., Yosinski, J., and Farhadi, A. Supermasks in Superposition. *In Proc. of NeurIPS*, 2020.

[5] Ke, Z., Liu, B., and Huang, X. Continual Learning of a Mixed Sequence of Similar and Dissimilar Tasks. *In Proc. of NeurIPS*, 2020.

[6] Prakhar Kaushik, Alex Gain, Adam Kortylewski, and Alan Yuille. Understanding catastrophic forgetting and remembering in continual learning with optimal relevance mapping. *arXiv preprint arXiv:2102.11343*, 2021.

[7] French, R. M. Using semi-distributed representations to overcome catastrophic forgetting in connectionist networks. 1991

[8] Kirkpatrick, J., Pascanu, R., Rabinowitz, N., Veness, J., Desjardins, G., Rusu, A. A., Milan, K., Quan, J., Ramalho, T., Grabska-Barwinska, A., et al. Overcoming catastrophic forgetting in neural networks. *Proceedings of the national academy of sciences, 114(13):3521–3526*, 2017.

[9] Sharkey, N. E. and Sharkey, A. J. C. Backpropagation discrimination geometric analysis interference memory modelling neural nets. *Connection Science, 7(3-4):301– 330*, 1995b.

[10] Robins, A. Catastrophic forgetting in neural networks: the role of rehearsal mechanisms. *In Proceedings 1993 The First New Zealand International Two-Stream Conference on Artificial Neural Networks and Expert Systems*, pp. 65–68, Dunedin, New Zealand, 1993. IEEE Comput. Soc. Press. ISBN 978-0-8186-4260-9. doi: 10.1109/ ANNES.1993.323080. URL http://ieeexplore. ieee.org/document/323080/.