Do solutions for catastrophic forgetting degrade model robustness to naturalistic corruption and adversarial attacks?

CMSC 35200 Project Mid Report

Tuan Pham
tuanph18@uchicago.edu

Amit Pradhan
pradhanak@uchicago.edu

Sebastian Barrios
sebastian.barrios@chicagobooth.edu

Introduction

Developing a deep learning model, for example in image classification, is not only about achieving the best classification accuracy. Even with increasing computational power, there are multiple issues that also need to be solved, including - to name a few - fast inference time, manageable memory size, continual learning and robustness to data/model corruption. An attractive solution for fast computation, efficient power consumption and potentially easier hardware implementation is binary neural networks (BNN) [1], in which only the signs of the model's hidden weights are utilized during inference.

A recent study takes inspiration from biological metaplasticity to solve catastrophic forgetting (CF) problems and continual learning problems for BNN [2]. More specifically, by creating a form of multiplicative gating during learning for hidden weights to represent weight consolidation, they are able to solve the permuted-MNIST task, sequential learning with CIFAR-10/100 dataset, and stream learning with these datasets. This approach shares certain similarities with another study in regular neural networks, also inspired by neuroscience literature, which takes into account (a) weight stabilization based on task importance, combined with (b) context-dependent gating allowing for more sparse, non-overlapping population activations, to facilitate learning and remembering a large number of tasks [3].

Back to BNNs, due to such quantization, they can be quite sensitive to corruption, for example data adversarial attacks [4] or potential soft errors in hardware accelerators [5]. Hence, we wish to ask whether previous solutions for CF [2] could further degrade model robustness to data corruption or help ameliorate it.

In conclusion, we propose a plan to (i) replicate the selected studies that tackle the topic of catastrophic forgetting, and (ii) observe whether solutions to avoid CF lead to models that have a higher level of degradation when different types of naturalistic corruption to input data are introduced [6], as well as the potential resulting vulnerability of these models to adversarial attacks [4].

Methods

The specific methods will be discussed in detail later. Code repository currently at:

https://github.com/tuanpham96/bnn-cf-vs-robust

Texts in blue are our next steps, or planned steps.

(*Initial*) Results

Experiment 0: *Replication of BNN training with metaplasticity parameter with permuted MNIST*

Before applying corruptions to the dataset, we first set out to observe whether the original results from [2] were replicabled. So we ran their codes for the multipercepton (MLP) version of the BNN with 2 hidden layers (1024 units each) for different values of the $meta$ hyperparameters: low $meta$ means binary hidden weights are allowed to change easily while high means such weights' own plasticity is constrained by their magnitude. This is shown in **Fig. 1**b. Consistent with the paper, higher $meta$ leads to better performance (potentially criticality is around 1.0), in which earlier tasks still perform well. Interestingly, too high value 2.0) might lead to performing not as well (though still good) for the current task - this is basically because high values would bias most changes happening for earlier tasks and would require longer time to learn the current task. Regardless, these results confirm the benefits of the metaplasticity to solve the permuted MNIST task, which is one of the benchmarks for catastrophic forgetting.

Additionally, we also try to replicate the relationship between the sizes of the hidden layers and the

performance (**Fig. 1**c). Contrary to the monotonic trend between the trend and the performance shown in the original paper, we found that large networks could be harmful. For now, we are unsure why. One of the reasons might be that we only allowed for 20 epochs per task for this benchmark while the original paper allowed for 40 epochs. Maybe larger networks require a bit more time to consolidate properly. We might run our training again with the exact same hyperparameter settings to see whether it might just have been some nuances in the settings.

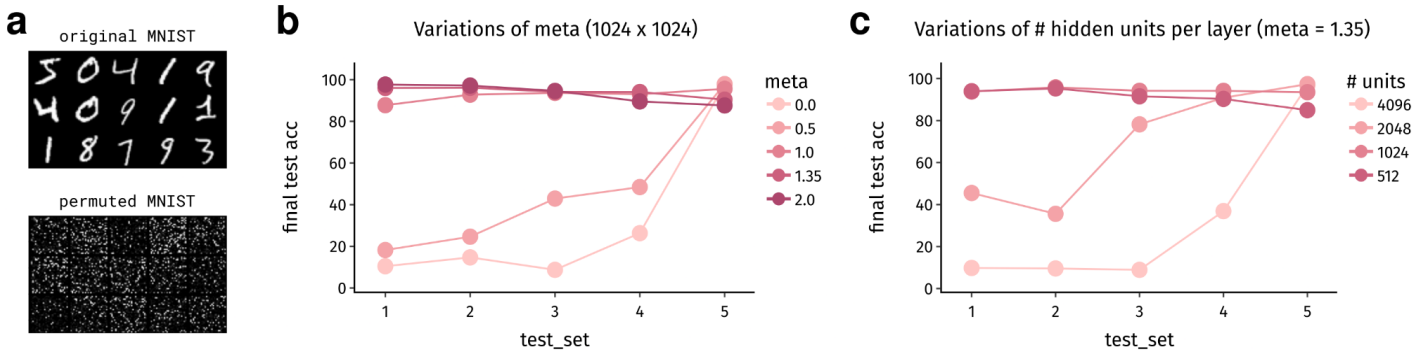

Figure 1 *Results of permuted MNIST tasks for the BNNs with metaplasticity training.* (**a**) Example of original (top, task 1) and generated permuted (bottom, after task 1) MNIST dataset. (**b**) Performance of the BNN (MLP, 2 hidden layers, 1024 units each) with different values of metaplasticity hyperparameter (**c**) Performance with variations in the number of hidden units per layer with $meta = 1.35$

Experiment 1.1: ***Metaplasticity-trained BNN in response to natural corruptions to pMNIST tasks***

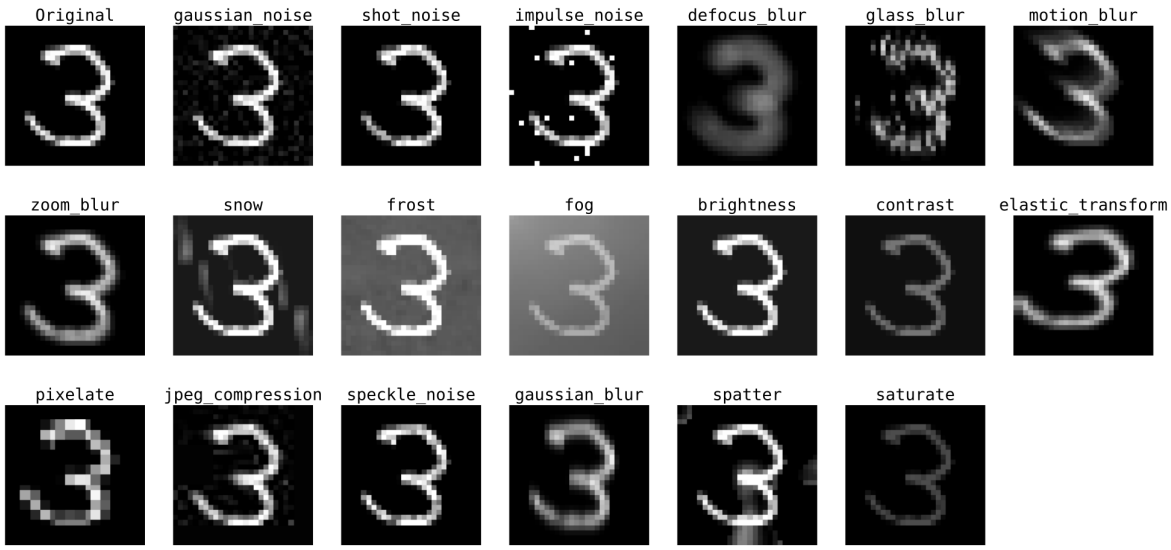

Figure 2 *Examples of the natural corruptions for MNIST data, plotted with similar color limits*

Next we moved on to applying perturbations onto our MNIST and pMNIST datasets after training the BNN models with metaplasticity optimization. First, we modified the code from [6] to allow for monochrome natural corruptions on our (p)MNIST datasets. Because some of these transformations are not easily standardized for tensor transformation, these corrupted data sets are pre-generated before training, to save time during testing. The examples of these corruption transformations are shown in **Fig. 2**. It must be noted that these types of corruption are possibly more realistic when considered with colored images like CIFAR or ImageNet, and the more appropriate benchmarks for catastrophic forgetting coupled with these types of perturbations would be MNIST-FashMNIST or split-class CIFAR. Regardless, for now we chose permuted MNIST as our first approach to assess catastrophic forgetting with natural corruptions.

The results are shown in **Fig. 3** for the models at the end of training for different values of $meta$, tested for the

original (last panel) test data sets of all tasks (i.e. *task data source*) and the natural corruptions shown in **Fig. 2**. For many different perturbations (for example *brightness* or *shot_noise, impulse_noise*), at least the inclusion of the metaplasticity training seems to help safeguard the model against the corruption for earlier tasks, though there's not a consistent trend for the actual magnitude of $meta$. At the same time, for many of these transformations, no improvement is observed regardless. It must be noted that these models are 2048 x 2048, in which **Fig. 1**c already shows that this particular setting is not optimal. Hence, we would need to rerun this with more variations of the sizes just to be certain.

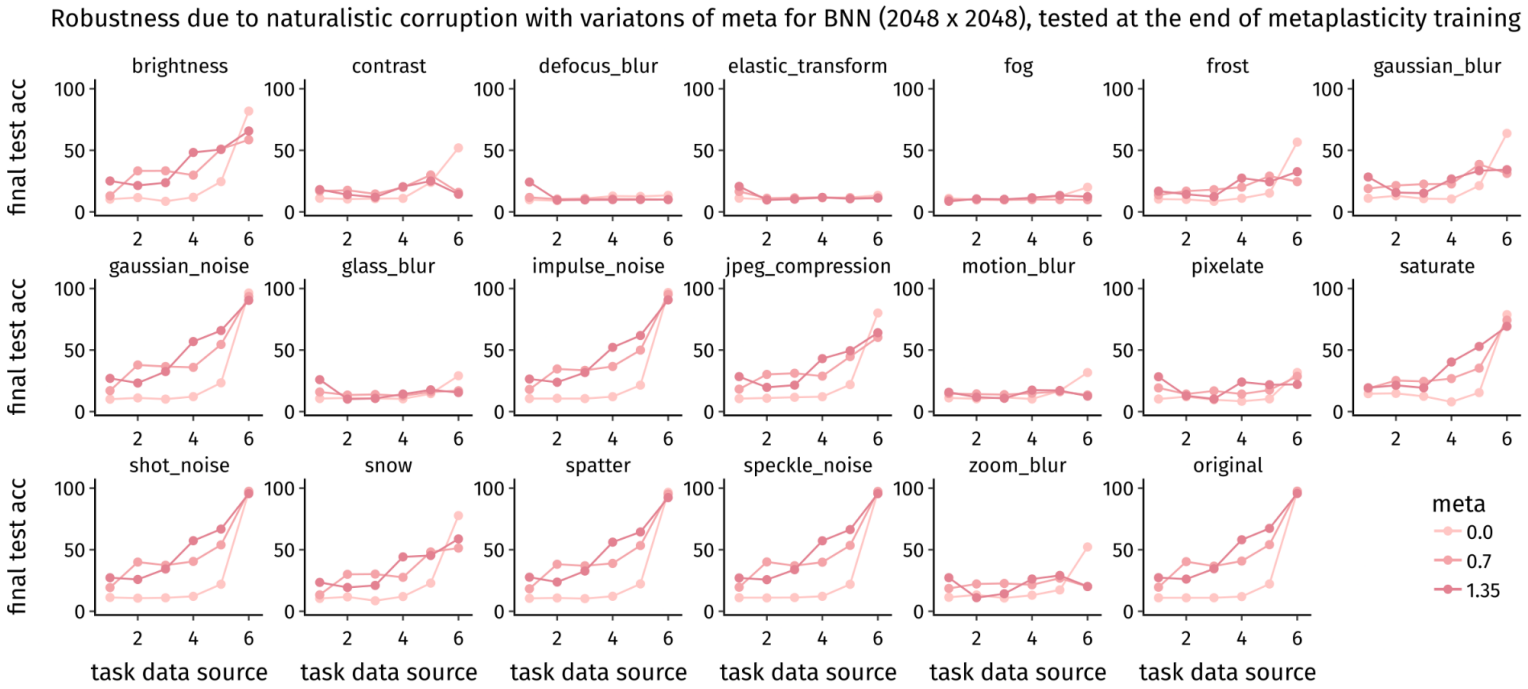

Figure 3 *Test performance of the BNN models at the end of metaplasticity (colors) training in response to different monochrome natural corruptions (panels, original is last) generated from different tasks.* The model here is MLP BNN with 2 hidden layers of 2048 units each. See **Fig 2.** for demonstrations of natural corruption. The x-axis for each panel represents where the original test data come from, e.g. *task-2* in the *shot_noise* panel means that the test dataset is the *task-2* (permuted MNIST) that gets corrupted with *shot_noise* corruption.

Experiment 1.2: *Metaplasticity-trained BNN in response to adversarial attacks to original MNIST data*

We have reproduced the effectiveness of a white-box adversarial attack '*Fast Gradient Sign Attack* (FGSM)' with the goal of misclassification on BNN. Adversarial attacks in general try to add the least amount of perturbations to the input data to cause the desired misclassification. FGSM is a very powerful yet intuitive attack. For a given input, FGSM calculates the gradients of the loss function with respect to the input and adjusts the input by adding a small perturbation in the directions of the gradients to cause the misclassification.

Currently, we have only examined the attacks on the original MNIST dataset (i.e. *task-01*) with the models trained like above. The results are shown in **Fig. 4** (some examples are shown in **Fig. 5**), illustrating that BNN is quite sensitive to adversarial attacks and continual learning worsens the performance significantly. The metaplasticity process does not seem to lead to much improvement to guard against FGSM attacks, potentially only very little for quite small perturbation strengths. However, we are unclear yet how accuracy could reach below chance here. Regardless, we would train the models again for appropriate sizes, as well as exploring smaller perturbation strengths (< 0.05) to see whether there are actually clear benefits even in this range. Additionally, we could consider integrating the Lipschitz regularization term for the hidden weight matrices, as proposed in [4], in addition to metaplasticity training to safeguard against such attacks; then we could test again for both catastrophic forgetting, naturalistic corruption and adversarial attacks.

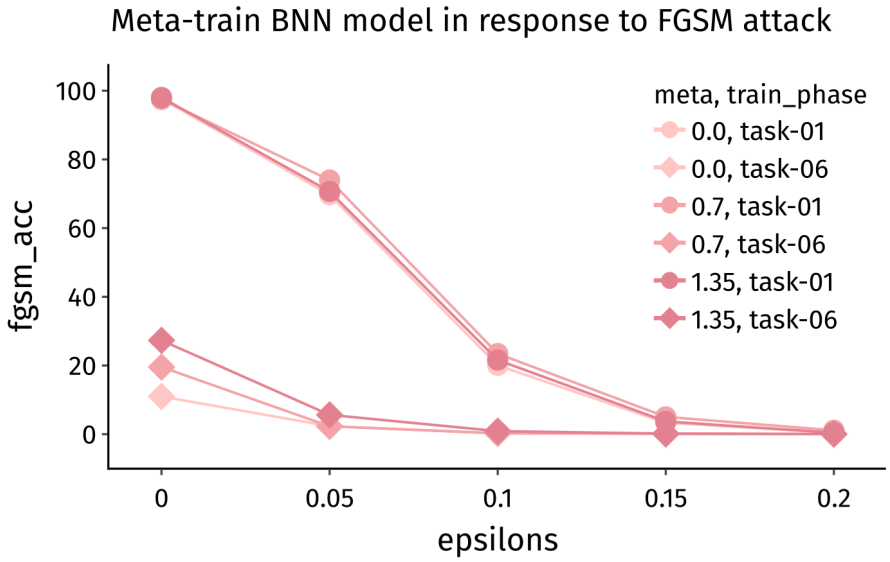

Figure 4 *FGSM adversarial attack on MNIST data of the BNN trained with metaplasticity during permuted MNIST task for 2 different training phases of the model*. The test accuracy due to FGSM attacks is shown as a function of the perturbation strengths *epsilons*. Trained at *task-01* refers to the end of the training phase in which the model was trained at MNIST dataset, while *task-06* means that the model was the end of the entire metaplasticity training process. Only attacks on the original MNIST were tested here. Future iterations will integrate attacks on the following tasks (i.e. on pMNIST).

Potential next steps

In addition to (blue texts) re-running some of the trainings with a few more variations and re-testing on both natural corruptions, as well as adversarial attacks with FGSM with potential integration of Lipschitz regularization, here are a few other things we need to consider:

(1) Since the corruptions are more realistic for datasets such as CIFAR, we might need to start integrating the split-CIFAR with convolutional BNN, then re-apply these robustness tests.
(2) Measuring training times, memory and energy for the permuted MNIST tasks.
(3) Run one of the continuous-valued networks in the paper to have something to compare (2) with.

References

[1] Courbariaux, Matthieu, Itay Hubara, Daniel Soudry, Ran El-Yaniv, and Yoshua Bengio. 2016. "Binarized Neural Networks: Training Deep Neural Networks with Weights and Activations Constrained to +1 or -1." *arXiv [cs.LG]*. arXiv. http://arxiv.org/abs/1602.02830.

[2] Laborieux, Axel, Maxence Ernoult, Tifenn Hirtzlin, and Damien Querlioz. 2021. "Synaptic Metaplasticity in Binarized Neural Networks." *Nature Communications* 12 (1): 2549.

[3] Masse, Nicolas Y., Gregory D. Grant, and David J. Freedman. 2018. "Alleviating Catastrophic Forgetting Using Context-Dependent Gating and Synaptic Stabilization." *arXiv [cs.LG]*. arXiv. http://arxiv.org/abs/1802.01569.

[4] Lin, Ji, Chuang Gan, and Song Han. 2019. "Defensive Quantization: When Efficiency Meets Robustness." *arXiv [cs.LG]*. arXiv. http://arxiv.org/abs/1904.08444.

[5] Khoshavi, Navid, Connor Broyles, and Yu Bi. 2020. "Compression or Corruption? A Study on the Effects of Transient Faults on BNN Inference Accelerators." In *2020 21st International Symposium on Quality Electronic Design (ISQED)*, 99–104.

[6] Hendrycks, Dan, and Thomas G. Dietterich. 2018. "Benchmarking Neural Network Robustness to Common Corruptions and Surface Variations." *arXiv [cs.LG]*. arXiv. http://arxiv.org/abs/1807.01697.

[7] Leavitt, Matthew L., and Ari S. Morcos. 2020. "On the Relationship between Class Selectivity, Dimensionality,

and Robustness." *arXiv [cs.LG]*. arXiv. http://arxiv.org/abs/2007.04440.

[8] Nathan Inkawhich, ADVERSARIAL EXAMPLE GENERATION, https://pytorch.org/tutorials/beginner/fgsm_tutorial.html#adversarial-example-generation

Figure 5 *The input images for different values of epsilon*. This is for FGSM attack on MNIST dataset for model trained at the end of the *task-01* (i.e. original MNIST) data without any metaplasticity.