MINIMIZING CHANGE IN CLASSIFIER LIKELIHOOD TO MITIGATE CATASTROPHIC FORGETTING

Anonymous authors
Paper under double-blind review

ABSTRACT

Continual learning is a longstanding goal of artificial intelligence, but is often confounded by "catastrophic forgetting" that prevents neural networks from learning tasks sequentially. Previous methods in continual learning have demonstrated how to mitigate catastrophic forgetting, and learn new tasks while retaining performance on the previous tasks. We analyze catastrophic forgetting from the perspective of change in classifier likelihood and propose a simple L_1 minimization criterion that can be adapted to different use cases. We further investigate two ways to minimize forgetting, as quantified by this criterion, and propose strategies to achieve finer control over forgetting. Finally, we evaluate our strategies on three datasets of varying difficulty, and demonstrate improvements over previously known L_2 strategies for mitigating catastrophic forgetting.

1 INTRODUCTION

Machine learning has achieved successes in many applications, including image recognition, game-playing, content recommendation and health-care (LeCun et al., 2015). Most of these systems require large amounts of training data and careful selection of architecture and parameters. Moreover, such systems often have to adapt to changing real-world requirements, and therefore changes in the data. Under these circumstances it is usually desired to retain performance on previous data while learning to perform well on training data with a different distribution. This is what constitutes continual learning (McCloskey, 1989).

A well known problem in the context of continual learning is "catastrophic forgetting" (Goodfellow et al., 2013), which occurs when the training process ends up modifying weights crucial to the performance on the previous data.

There has been a lot of work in trying to overcome catastrophic forgetting. Broadly, the approaches in the literature try to mitigate forgetting in three ways: (a) *architectural* approaches (Yoon et al., 2018; Li et al., 2019) try to incrementally grow the network to learn the new task through added capacity, (b) *regularization* approaches (Kirkpatrick et al., 2016; Zenke et al., 2017; Wiewel & Yang, 2019) regularize changes to crucial weights, so that the network can learn to perform well on the new task while preserving the performance on the previous tasks (assuming the network has enough capacity for all tasks), and (c) *memory* approaches (Lopez-Paz, 2017; Nguyen et al., 2018) store examples from each task being learned and then learn a new task while simultaneously maximizing performance on each of the stored memories.

Performance in these works is often judged with respect to overall accuracy. In the present work, we specifically consider exactly what has been forgotten and what has been learned. Such considerations may be important in safety-critical systems or in systems that have been calibrated. For example, in safety-critical systems, it may not be acceptable to maintain overall performance by trading validated decisions for correct decisions that have not been validated. Likewise, the calibration of a system may require that all decisions, good and bad, remain the same.

For the purposes of this paper, we focus on regularization strategies. Regularization strategies typically formulate continual learning in two ways: (a) from a Bayesian perspective (Kirkpatrick et al., 2016; Lee et al., 2017; Liu et al., 2018; Chaudhry et al., 2018) where the goal is to learn the newest task while simultaneously minimizing the KL-divergence between the posterior log likelihood distribution and the prior (see Section 2), or (b) by trying to minimize large changes to influential weights

for previous tasks (Zenke et al., 2017; Wiewel & Yang, 2019). Both these formulations produce an L_2 regularization objective and mitigate forgetting by penalizing changes to weights important to task performances. However, their exact effect on change in classifier likelihood is not known. In this paper, we attempt to quantify this change in classifier likelihood more directly and then use it to provide a generic criterion that can be adapted to different use cases of likelihood preservation.

Our contributions are as follows: we propose a more general framework to mitigate catastrophic forgetting, which involves directly penalizing the change in the classifier likelihood functions. Specifically: (a) we analyze catastrophic forgetting and provide a generic L_1 minimization criterion to mitigate it, (b) we propose two strategies to utilize this criteria and discuss how the cross-entropy loss can be reformulated to achieve finer control over forgetting, and (c) we evaluate these strategies on three datasets and demonstrate improvements over traditional L_2 regularization strategies like elastic weight consolidation (EWC) (Kirkpatrick et al., 2016) and synaptic intelligence (SI) (Zenke et al., 2017).

2 Background of Continual Learning

Formally, let the tasks correspond to datasets $D_1, D_2, \cdots, D_n$ such that the goal in task i is to achieve the maximum performance on a dataset $D_i = (X_i, Y_i) = (\{x_i^{(k)}\}_{k=1}^{K_i}, \{y_i^{(k)}\}_{k=1}^{K_i})$, which has K_i examples. Let the likelihood function be approximated by a ReLU feedforward neural network (final layer is followed by softmax) with weights θ, that is, given an example x, the network produces a set of probabilities $\{P_\theta(y = j|x)\}_{j=1}^{M}$, where M is the number of classes. For notational simplicity, we denote $P_\theta(y = j|x)$ as $P_\theta^j(\cdot|x)$. If the ground truth for x is g, where $(1 \leq g \leq M)$, then we use the shorthand for the predicted likelihood of g as $P_\theta(\cdot|x) := P_\theta^g(\cdot|x)$.

For any task i, minimizing its specific cross entropy loss $\mathcal{L}_i(\theta)$ achieves the best performance for task i, which can be written as:

$$\mathcal{L}_i(\theta) = -\sum_{k=1}^{K_i} \log P_\theta(\cdot|x_i^{(k)})$$

For any task i, the ideal weights achieved at the end of task i should also retain performances on tasks $1, 2, \cdots, i-1$. Therefore, ideally, the overall *joint* cross entropy loss over datasets $D_1, D_2, \ldots, D_i$ should be minimized:

$$\mathcal{L}_{1:i}(\theta) = -\sum_{j=1}^{i} \sum_{k=1}^{K_j} \log P_\theta(\cdot|x_j^{(k)})$$

Joint training quickly becomes expensive as the number of tasks grow, but has the best performance across all tasks that were trained on (Li & Hoiem, 2017).

Bayesian continual learning formulates learning a new task as trying to maximize a posterior $p(\theta|D_{1:i})$ given a prior $p(\theta|D_{1:i-1})$. So, if the weights θ at the end of task $i-1$ and i are denoted by $\theta^*_{1:i-1}$ and $\theta^*_{1:i}$ respectively, then the prior and the posterior can be thought of as the predicted likelihood distributions represented by the neural network at $\theta^*_{1:i-1}$ and $\theta^*_{1:i}$. For every task i, the Bayesian formulation tries to minimize $\mathcal{L}_i(\theta)$ as well as the dissimilarity between the prior and the posterior.

EWC uses the KL-divergence of the two predicted likelihood distributions as the dissimilarity metric. Assuming the difference between $\theta^*_{1:i-1}$ and $\theta^*_{1:i}$ is small, the second order Taylor approximation of the KL-divergence produces (with a further diagonal approximation of the Fisher matrix):

$$D_{KL}\big(p(\theta^*_{1:i})||p(\theta^*_{1:i-1})\big) \approx \frac{1}{2}(\theta^*_{1:i} - \theta^*_{1:i-1})^2 \cdot \mathbf{E}_{(x,y)\sim D_{1:i-1}}\left[\Big(\nabla_\theta \log P_\theta(\cdot|x)\Big)^2\Big|_{\theta=\theta^*_{1:i-1}}\right]$$

For any task $i \geq 2$, EWC minimizes the sum of $\mathcal{L}_i(\theta)$ and this approximation (multiplied by a $\lambda \geq 0$). λ acts as a hyperparameter that controls the weight of the penalty for a specific learned task. It is typically kept the same for all learned tasks.

3 ANALYSIS OF THE CHANGE IN LIKELIHOOD

After learning a task i, the weights of the network are $\theta^*_{1:i}$. For simplicity, let $\theta^*_{1:i} \equiv \theta^*$ and that afterwards, at any point in the sequential training process, the weights are at $\theta^* + \Delta\theta$. Assuming $\Delta\theta$ is small, we can apply a first order Taylor approximation of the individual predicted likelihood $P^j_{\theta^*+\Delta\theta}$ in the neighbourhood of $\theta = \theta^*$:

$$P^j_{\theta^*+\Delta\theta} \approx P^j_{\theta^*} + \Delta\theta \cdot (\nabla_\theta P^j_\theta)|_{\theta=\theta^*}$$

The individual predicted likelihood P^j_θ on an example $x \in D_i$ changes by the magnitude $|P^j_{\theta^*+\Delta\theta}(\cdot|x) - P^j_{\theta^*}(\cdot|x)|$. The average magnitude change in P^j_θ over the dataset D_i is given by the expectation:

$$\begin{aligned}\mathbf{E}_{(x,y)\sim D_i}\left[\left|P^j_{\theta^*+\Delta\theta}(\cdot|x) - P^j_{\theta^*}(\cdot|x)\right|\right] &\approx \mathbf{E}_{(x,y)\sim D_i}\left[\left|\Delta\theta \cdot (\nabla_\theta P^j_\theta(\cdot|x)|_{\theta=\theta^*})\right|\right] \\ &\leq \|\Delta\theta\|_1 \cdot \mathbf{E}_{(x,y)\sim D_i}\left[\left\|\nabla_\theta P^j_\theta(\cdot|x)\big|_{\theta=\theta^*}\right\|_1\right]. \qquad (1)\end{aligned}$$

At every task i, we can minimize directly the average change in predicted likelihood for the previous datasets, and this minimization should mitigate catastrophic forgetting. This constitutes our *minimization criterion*. Depending on the requirement, the minimization criterion can be interpreted to provide a regularization objective. In this paper, we identify four broad use cases of this criterion:

Case I. We can preserve the entire set of predicted likelihoods from θ^* to $\theta^* + \Delta\theta$, which would penalize changes to any individual predicted likelihood. This is the most restrictive version of the criterion and can be achieved by regularizing a sum over $j = 1$ to M of the individual changes.

Case II. We can preserve the change in predicted likelihood for the predicted label at θ^*, which corresponds to the highest individual probability in $\{P^j_{\theta^*}\}_{j=1}^M$. This may be desired in tasks related to safety-critical systems (e.g., autonomous driving), where a network has been safety-calibrated at deployment and now needs to add some more knowledge without violating previously satisfied safety constraints.

To achieve this, we can use the expectation over $\left|(P^j_{\theta^*+\Delta\theta}(\cdot|x) - P^j_{\theta^*}(\cdot|x)) \cdot P^j_{\theta^*}(\cdot|x)\right|$ rather than the original formulation in (1) and then regularize a sum over $j = 1$ to M like in Case I. In most cases, this term would evaluate to the difference in the individual predicted likelihoods for the predicted label at θ^* (since the probabilities are output by a softmax layer).

Case III. We can preserve the change in predicted likelihood for the ground truth by computing the expectation for $\left|P_{\theta^*+\Delta\theta}(\cdot|x) - P_{\theta^*}(\cdot|x)\right|$.

Case IV. We can partially preserve the change in predicted likelihood for the ground truth, that is, penalize the change $P_{\theta^*}(\cdot|x) = 1 \rightarrow P_{\theta^*+\Delta\theta}(\cdot|x) = 0$ but allow the change $P_{\theta^*}(\cdot|x) = 0 \rightarrow P_{\theta^*+\Delta\theta}(\cdot|x) = 1$ for the ground truth predicted likelihood. This applies the penalty only when a correctly classified x at θ^* becomes incorrectly classified at $\theta^* + \Delta\theta$. The expectation is then computed over $\left|(P_{\theta^*+\Delta\theta}(\cdot|x) - P_{\theta^*}(\cdot|x)) \cdot P_{\theta^*}(\cdot|x)\right|$, similar to Case II.

In all of these cases, we end up with a direct minimization criterion that can be minimized just like in EWC. In fact, the quadratic loss penalty proposed in Kirkpatrick et al. (2016), which was later corrected in Huszár (2018) to more appropriately represent Bayesian updating, can be interpreted as the upper-bound of the squared L_2 version of the change in predicted likelihood as described in Case III.

$$\begin{aligned}\mathbf{E}_{(x,y)\sim D_i}\left[\left(P_{\theta^*+\Delta\theta}(\cdot|x) - P_{\theta^*}(\cdot|x)\right)^2\right] &\approx \mathbf{E}_{(x,y)\sim D_i}\left[\left(\Delta\theta \cdot (\nabla_\theta P_\theta(\cdot|x)|_{\theta=\theta^*}\right)^2\right] \\ &\leq \mathbf{E}_{(x,y)\sim D_i}\left[\|\Delta\theta\|_2^2 \cdot \left\|\nabla_\theta P_\theta(\cdot|x)|_{\theta=\theta^*}\right\|_2^2\right] \\ &= \|\Delta\theta\|_2^2 \cdot \mathbf{E}_{(x,y)\sim D_i}\left[\left\|\left(P_\theta(\cdot|x) \cdot \nabla_\theta \log P_\theta(\cdot|x)\right)\big|_{\theta=\theta^*}\right\|_2^2\right] \\ &\leq \|\Delta\theta\|_2^2 \cdot \mathbf{E}_{(x,y)\sim D_i}\left[\left\|\nabla_\theta \log P_\theta(\cdot|x)|_{\theta=\theta^*}\right\|_2^2\right]\end{aligned}$$

Intuitively, therefore, the quadratic loss penalty works even when not computed as specified in Huszár (2018) because it penalizes the upper bound of the squared L_2 change in likelihood for task i.

4 DIRECT MINIMIZATION OF THE EXPECTED CHANGE IN LIKELIHOOD

Given (1), we propose two strategies to minimize the expected change in predicted likelihood.

Method 1 (Soft Regularization). The upper bound in (1) can be directly regularized per task. With the L_1 change, the loss per task i becomes:

$$\mathcal{L}_i(\theta) + \lambda' \cdot \sum_{k=1}^{i-1} \left\|\theta - \theta^*_{1:k}\right\|_1 \cdot \mathbf{E}_{(x,y)\sim D_k}\left[\left\|\nabla_\theta P_\theta(\cdot|x)\big|_{\theta=\theta^*_{1:k}}\right\|_1\right] \tag{2}$$

L_1 regularization is known to produce sparser solutions than L_2 regularization above a critical λ, that is, requiring fewer non zero weights, in the context of plain weight regularization (Moore & DeNero, 2011). This is because the L_1 objective penalizes change in weights more strongly than in L_2 and forces the weights to stay closer to 0. *In the context of predicted likelihood preservation*, similarly it should be expected that the L_1 penalty penalizes change in predicted likelihoods more strongly than L_2 and forces the change in likelihoods to stay closer to 0.

With the 4 cases described in Section 3, Method 1 can be used in 4 ways. We denote these 4 methods as DM-I, DM-II, DM-III and DM-IV respectively, where DM stands for "Direct Minimization".

Constrained learning. Better preservation of predicted likelihood also has a downside, that is, if the previous behaviour is preserved too much, the network is unable to learn the new task well. To counteract this, we introduce two parameters - c_1 and c_2, to constrain the learning. For notational simplicity, let us denote the expectation in (2) as $G(\theta^*_{1:k}, D_k)$.

After task i has been learned, the absolute change in predicted likelihood (for the use case) is upper bounded by $\|\theta - \theta^*_{1:i}\|_1 \cdot G(\theta^*_{1:i}, D_i)$. We can turn off the training on the cross entropy loss after the upper bound on absolute change in likelihood is $\geq c$. This can be achieved by modifying the MLE loss to be:

$$\mathcal{L}_i(\theta) \leftarrow \mathcal{L}_i(\theta) \cdot \prod_{k=1}^{i-1} \frac{1}{2}\Big\{1 + \text{sign}\big(c_k - \|\theta - \theta^*_{1:k}\|_1 \cdot G(\theta^*_{1:k}, D_k)\big)\Big\} \tag{3}$$

In fact, it is more advantageous to maintain a moving c_i for every task i which is initialized with $c_i \leftarrow c_1$, and then increased to $c_i \leftarrow c_i + c_2$ after every new task ($c_1, c_2 \geq 0$). This kind of thresholding provides a direct way to bound the amount of forgetting, compared to the unconstrained learning in EWC or SI. The advantage of this kind of thresholding is evident from our experiments (Table 3).

Method 2 (Freezing). With any soft regularization strategy, all the weights are always updated, even if the changes to some weights are very small. This might perturb sensitive weights, even if by a small amount. These small perturbations can add up over multiple tasks and eventually end up affecting the classifier likelihood irreversibly.

The upper bound of change in classifier likelihood for a dataset D_i is dependent on two terms (see (1)), $\|\Delta\theta\|_1$ and the expectation of the norm of the gradients. To minimize the change in classifier likelihood, we may opt to minimize $\|\Delta\theta\|_1$ more conventionally, by freezing the most important weights. This reduces the magnitude of $\Delta\theta$ and therefore results in a lesser change in likelihood. Other strategies in the literature have tried similar approaches (for eg. Serra et al. (2018)).

To assess the effects of this kind of freezing separately from L_1 criterion, we freeze weights on EWC. We denote this method as DM-p. Specifically, the Fisher information matrix already contains information about which parameters are important, and should not be disturbed. We impair the gradient update by setting the gradients of top $p\%$ important parameters to 0 for each task $i \geq 2$.

Table 1: Best hyperparams ($\eta = 0.0001, h = 128$).

Dataset	λ (EWC)	λ'	c_1, c_2				p	c, ζ (SI)
			I	II	III	IV		
P-MNIST	10^1	1	0.04,0.02	0.02,0.04	0.02,0.0	0.1,0.08	0.4	0.01,0.1
S-MNIST	10^4	10^1	0.02,0.0	0.02,0.1	0.04,0.04	0.02,0.02	0.4	2,0.001
Sim-EMNIST	10^4	1	0.04,0.02	0.02,0.1	0.04,0.08	0.04,0.02	0.2	2,0.001

Table 2: Mean (std) of the final average accuracy (%) with the best hyperparameters, 5 seeds. Only the best result from DM-I, II, III, IV (constrained as well as unconstrained) is shown. The comparison among the constrained and unconstrained variants of DM are given in Table 3.

Dataset	Baseline	EWC	SI	DM-p	DM (best)
P-MNIST	54.95 (2.07)	93.91 (0.51)	93.84 (0.49)	94.47 (0.26)	**95.13** (0.24)
S-MNIST	63.11 (0.53)	69.28 (3.69)	78.57 (2.25)	72.02 (2.80)	**80.12** (0.51)
Sim-EMNIST	75.48 (1.26)	89.69 (3.10)	90.67 (1.99)	89.15 (2.95)	**91.93** (1.77)

Table 3: Mean (std) of the final average accuracy (%) with the best hyperparameters for DM-I, II, III, IV, constrained (top) and unconstrained (bottom), 5 seeds. Values are also provided for EWC for comparison.

Dataset	EWC (L_2)	DM-I	DM-II	DM-III	DM-IV
P-MNIST	93.91 (0.51)	94.73 (0.33)	94.19 (0.25)	95.02 (0.28)	**95.13** (0.24)
S-MNIST	69.28 (3.69)	77.24 (1.96)	77.56 (2.36)	80.09 (1.65)	**80.12** (0.51)
Sim-EMNIST	89.69 (3.10)	90.17 (1.31)	91.88 (1.07)	91.66 (0.98)	**91.93** (1.77)

Dataset	EWC (L_2)	DM-I	DM-II	DM-III	DM-IV
P-MNIST	93.91 (0.51)	93.33 (0.61)	90.13 (1.16)	**95.01** (0.27)	94.99 (0.30)
S-MNIST	69.28 (3.69)	**77.24** (1.96)	76.61 (0.96)	77.06 (1.83)	77.01 (1.67)
Sim-EMNIST	**89.69** (3.10)	84.56 (0.66)	83.01 (1.14)	86.27 (1.07)	85.33 (0.73)

Table 4: Mean (std) of the likelihood retention R (%) with the best hyperparams for DM-I, II, III, IV (constrained) and DM-p, 5 seeds.

Dataset	EWC (L_2)	DM-I	DM-II	DM-III	DM-IV	DM-p
P-MNIST	95.46 (0.43)	97.25 (0.20)	97.07 (0.10)	**98.05** (0.37)	98.01 (0.22)	96.64 (0.31)
S-MNIST	79.17 (2.27)	72.53 (1.83)	71.94 (2.70)	73.31 (2.56)	73.10 (1.71)	**81.56** (2.91)
Sim-EMNIST	**93.26** (2.03)	89.32 (1.82)	91.08 (1.77)	92.16 (1.33)	91.63 (1.53)	93.18 (2.12)

5 EXPERIMENTS

In this section we describe the methodology and results of our experiments (Tables 1–4).

Evaluated Methods. To assess the performance of our strategies, we evaluate our proposed methods and compare its performance with other L_2 variants in continual learning literature. Following are the methods we evaluate:

- **Baseline**: Trained with just the likelihood loss (no regularization).
- **EWC**: Accumulated Fisher information matrices and combined quadratic losses, as described in Kirkpatrick et al. (2016); Huszár (2018); Kirkpatrick et al. (2018).
- **SI**: Synaptic Intelligence strategy as described in Zenke et al. (2017).
- **DM-I, II, III, IV**: Proposed in Section 4, soft regularization strategy (Method 1); 4 variants described in Section 3. For each variant, we conduct experiments with both constrained and unconstrained learning of L_1 criterion.
- **DM-p**: Freezing strategy (Method 2) described in Section 4; implemented on EWC.

Training methodology. Training is done on feedforward ReLU networks for each strategy with 2 hidden layers ($h = 128, \eta = 0.0001$) for 20 epochs. For hyperparameter search, we evaluate all methods on a single seed. Then the final results are reported across 5 seeds with the best parameter (mean and standard deviation are shown). All hyperparameters used are reported in Table 1. Table 2, 3 show the performance (accuracy) of the proposed methods. Additionally, we also assess the retention of predicted likelihood. To calculate the likelihood retention, we first compute the predictions per task after the task has been fully trained, then calculate how many of these predictions have changed at the end of future tasks. If the retained predictions for task i at the end of task j ($j > i$) is $R_{i,j}$, then we define likelihood retention after n tasks as (reported in Table 4):

$$R = \frac{1}{n-1} \sum_{j=2}^{n} \frac{1}{j-1} \left(\sum_{i=1}^{j-1} R_{i,j} \right)$$

Datasets. We evaluate on the following datasets:

- **Permuted MNIST**: 5 task version; every task is 10-class classification on the MNIST dataset with permuted pixels; used in Kirkpatrick et al. (2016); Zenke et al. (2017); Nguyen et al. (2018); Li et al. (2019).
- **Split MNIST**: 5 tasks where every task is 2-class classification; The tasks are labels 0/1, 2/3, 4/5, 6/7, 8/9 from the MNIST dataset; used in Chaudhry et al. (2018); Wiewel & Yang (2019).
- **Similar EMNIST**: Hand-picked labels from the EMNIST dataset such that the classification tasks are roughly similar; 4 tasks, 3-class classification, tasks are 2/O/U, Z/8/V, 7/9/W, T/Q/Y.

We use the Adam optimizer for our experiments. Constants searched for EWC include $\lambda = \{1, 10^1, 10^2, 10^3, 10^4\}$. In constrained DM-I, II, III, IV, we searched for $\lambda' = \{1, 10^1, 10^2\}$, $c_1 = \{0.02, 0.04, \cdots 0.10\}$ and $c_2 = \{0.0, 0.02, 0.04, \cdots 0.10\}$. For DM-p, we searched for $p = \{0.1, 0.2, 0.3, \cdots 0.9\}$. For SI, we searched for $c = \{0.01, 0.1, 0.5, 1, 2\}$ and $\zeta = \{0.001, 0.01, 0.1, 1\}$.

6 DISCUSSION

In this section we give further insights about our results.

Hyperparameter choice. As can be seen in Table 1, EWC often requires a high λ to remember previous tasks better. In contrast, the L_1 methods perform well even with a small λ'. This can be explained by the fact that minimization with an L_2 method contains a $(||\Delta\theta||_2)^2$ term instead of $(||\Delta\theta||_1)$. This means that the weights (which are typically quite small) are squared in the L_2 methods, which then requires a stronger λ to compensate for the squaring. So, L_1 methods require a hyperparameter search over a smaller range of values.

Degree of preservation. A higher p in DM-p has the same effect as as a low c_1, c_2 in constrained DM-I, II, III, IV. If c_1, c_2 are too low, then the training switches off very early, and likewise, if p is too high, the requisite weights never change enough to adapt to the newest task. For the datasets considered, we find that fixing $20 - 40\%$ of the weights typically works the best in DM-p.

Improvements over L_2 methods.

- *P-MNIST* and *Sim-EMNIST*: EWC and SI are already known to perform well on P-MNIST. In our experiments with the 5-task variant of P-MNIST, they reach an average final accuracy of $\sim 94\%$. All of DM-I, DM-II, DM-III, DM-IV and DM-p outperform EWC and SI on the 5 task P-MNIST for the same number of epochs, as evidenced by Table 2. A large improvement was not expected, since EWC already performs well on these datasets.
- *S-MNIST*: S-MNIST is a difficult dataset because it only involves 2-class classification for each task, which means that the decision boundary found by the network at each task is

very susceptible to change in the decision boundary at the next task. This is why EWC is unable to reach beyond $\sim 69\%$ on S-MNIST. DM-p improves on this by a few points, but DM-I, II, III, IV all improve on EWC by $\sim 7 - 10\%$.

Effect of constrained learning. As can be seen in Table 3, tuned constrained DM-I, II, III, IV all perform better or similar than the tuned unconstrained counterparts.

Effect of different types of preservation on performance. While DM-I, II might be suited to specific applications, DM-III, IV typically perform the best in terms of accuracy improvement. This is expected, since DM-III, IV directly regularize change in predicted likelihood for the ground truth.

Effect of different types of preservation on retention. We observe mixed results with respect to retention. While it is expected that a higher retention should correspond to a lower amount of forgetting, Table 4 does not show that the L_1 criterion universally has the best retention across the tested datasets. Specifically, the retention advantage of the L_1 criterion is clear for P-MNIST, but it is not as clear for S-MNIST or Sim-EMNIST.

We speculate that this is because of the λ chosen for S-MNIST and Sim-EMNIST during the hyperparameter search. During the search, λ is optimized for the best accuracy. In order for EWC to have the best accuracy for these datasets (S-MNIST, Sim-EMNIST), the required hyperparameter λ is huge (10^4), which leads to an over-preservation of past classifier likelihood at the expense of the learning the likelihood for the newest task, while the proposed DM strategies use a normal λ' for their corresponding best performance. In fact, the huge λ leads to sub-optimal performance for the newest task in EWC, but maximizes the average final accuracy. The retention metric does not capture this sub-optimal performance.

Out of DM-I, II, III, IV, the method DM-III retains the most amount of predictions, empirically. For DM-p, the retention advantage is clearly better than plain EWC for P-MNIST and S-MNIST, and close to plain EWC for Sim-EMNIST.

7 Conclusions

Most real-world classification systems rely on connectionist networks, which are known to suffer from catastrophic forgetting when subjected to sequential learning tasks. Existing (regularization) strategies to mitigate catastrophic forgetting typically minimize an L_2 criterion, which can produce non-sparse solutions and require a costly hyperparameter search for the appropriate penalty weight.

In this paper, we proposed a more general criterion that involves direct minimization of the change in classifier likelihood and explained how to adapt the criterion to four broad use cases. Using this criterion, we identified two ways to improve the classifier performance: (a) by directly soft-regularizing the change in classifier likelihood and (b) by freezing influential weights. Both of these perform better than, or at least similar to, existing L_2 strategies. We further discussed the effect of various proposed classifier likelihood preservation methods and showed that preserving the classifier likelihood with respect to the ground truth is a good strategy to preserve classifier performance.

Future Work. Having compared our method to existing L_2 strategies, it would be interesting to compare and contrast the benefits and problems of the proposed L_1 strategies with other non-L_2 strategies for continual learning, e.g., IMM (Lee et al., 2017) and VCL (Nguyen et al., 2018). It would be also be interesting to see the effect of direct minimization strategies for more complicated and realistic image classification datasets, like CIFAR100 (Krizhevsky et al., 2009) and ImageNet (Deng et al., 2009).

References

Arslan Chaudhry, Puneet K Dokania, Thalaiyasingam Ajanthan, and Philip HS Torr. Riemannian walk for incremental learning: Understanding forgetting and intransigence. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 532–547, 2018.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pp. 248–255. Ieee, 2009.

Ian J. Goodfellow, Mehdi Mirza, Xia Da, Aaron C. Courville, and Yoshua Bengio. An empirical investigation of catastrophic forgeting in gradient-based neural networks. *CoRR*, abs/1312.6211, 2013.

Ferenc Huszár. Note on the quadratic penalties in elastic weight consolidation. *Proceedings of the National Academy of Sciences of the United States of America*, 115 11:E2496–E2497, 2018.

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. Overcoming catastrophic forgetting in neural networks. *Proceedings of the National Academy of Sciences of the United States of America*, 114 13:3521–3526, 2016.

James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, et al. Reply to huszár: The elastic weight consolidation penalty is empirically valid. *Proceedings of the National Academy of Sciences*, 115(11):E2498–E2498, 2018.

Alex Krizhevsky, Vinod Nair, and Geoffrey Hinton. Cifar-10 and cifar-100 datasets. *URl: https://www. cs. toronto. edu/kriz/cifar. html*, 6, 2009.

Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *nature*, 521(7553):436, 2015.

Sang-Woo Lee, Jin-Hwa Kim, Jaehyun Jun, Jung-Woo Ha, and Byoung-Tak Zhang. Overcoming catastrophic forgetting by incremental moment matching. In *Advances in neural information processing systems*, pp. 4652–4662, 2017.

Xilai Li, Yingbo Zhou, Tianfu Wu, Richard Socher, and Caiming Xiong. Learn to grow: A continual structure learning framework for overcoming catastrophic forgetting. In *International Conference on Machine Learning*, pp. 3925–3934, 2019.

Zhizhong Li and Derek Hoiem. Learning without forgetting. *IEEE transactions on pattern analysis and machine intelligence*, 40(12):2935–2947, 2017.

Xialei Liu, Marc Masana, Luis Herranz, Joost Van de Weijer, Antonio M Lopez, and Andrew D Bagdanov. Rotate your networks: Better weight consolidation and less catastrophic forgetting. In *2018 24th International Conference on Pattern Recognition (ICPR)*, pp. 2262–2268. IEEE, 2018.

David Lopez-Paz. Marc'aurelio ranzato. *Gradient episodic memory for continuum learning. NIPS*, 2017.

M. W. McCloskey. Catastrophic interference in connectionist networks: The sequential learning problem" the psychology. 1989.

Robert C. Moore and John DeNero. L1 and l2 regularization for multiclass hinge loss models. In *Symposium on Machine Learning in Speech and Natural Language Processing*, 2011. URL `http://www.ttic.edu/sigml/symposium2011/papers/Moore+DeNero_Regularization.pdf`.

Cuong V. Nguyen, Yingzhen Li, Thang D. Bui, and Richard E. Turner. Variational continual learning. In *International Conference on Learning Representations*, 2018. URL `https://openreview.net/forum?id=BkQqq0gRb`.

Joan Serra, Didac Suris, Marius Miron, and Alexandros Karatzoglou. Overcoming catastrophic forgetting with hard attention to the task. In *International Conference on Machine Learning*, pp. 4555–4564, 2018.

Felix Wiewel and Bin Yang. Localizing catastrophic forgetting in neural networks. *arXiv preprint arXiv:1906.02568*, 2019.

JaeHong Yoon, Jeongtae Lee, Eunho Yang, and Sung Ju Hwang. Lifelong learning with dynamically expandable network. In *International Conference on Learning Representations*. International Conference on Learning Representations, 2018.

Friedemann Zenke, Ben Poole, and Surya Ganguli. Continual learning through synaptic intelligence. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pp. 3987–3995. JMLR. org, 2017.