Gradient Episodic Memory for Continuum Learning

David Lopez-Paz and Marc'Aurelio Ranzato
Facebook Artificial Intelligence Research
{dlp,ranzato}@fb.com

Abstract

One major obstacle towards artificial intelligence is the poor ability of models to quickly solve new problems, without forgetting previously acquired knowledge. To better understand this issue, we study the problem of *learning over a continuum of data*, where the model observes, once and one by one, examples concerning an ordered sequence of tasks. First, we propose a set of metrics to evaluate models learning over a continuum of data. These metrics characterize models not only by their test accuracy, but also in terms of their ability to transfer knowledge across tasks. Second, we propose a model to learn over continuums of data, called Gradient of Episodic Memory (GEM), which alleviates forgetting while allowing beneficial transfer of knowledge to previous tasks. Our experiments on variants of the MNIST and CIFAR-100 datasets demonstrate the strong performance of GEM when compared to the state-of-the-art.

1 Introduction

The starting point in supervised learning is to collect a *training set* $D_{\text{tr}} = \{(x_i, y_i)\}_{i=1}^n$, where each *example* (x_i, y_i) is composed by a *feature vector* $x_i \in \mathcal{X}$, and a *target vector* $y_i \in \mathcal{Y}$. Most supervised learning methods assume that each example (x_i, y_i) is an identically and independently distributed (iid) sample drawn from a *fixed* probability distribution $P(X, Y)$, which describes a single learning task. The goal of supervised learning is to construct a model $f : \mathcal{X} \to \mathcal{Y}$, used to predict the target vectors y associated to unseen feature vectors x. To accomplish this, supervised learning methods often employ the Empirical Risk Minimization (ERM) principle [Vapnik, 1998], where f is found by minimizing $\frac{1}{|D_{\text{tr}}|} \sum_{(x_i,y_i) \in D_{\text{tr}}} \ell(f(x_i), y_i)$, where $\ell : \mathcal{Y} \times \mathcal{Y} \to [0, \infty)$ is a loss function penalizing prediction errors. In addition, ERM often requires multiple passes over the training set.

Unfortunately, ERM is a major simplification from what we deem as human learning. In stark contrast to learning machines, learning humans observe data as an ordered sequence, seldom observe the same example twice, they can only memorize a few pieces of data, and the sequence of examples concerns different learning tasks. Therefore, the iid assumption, along with any hope of employing the ERM principle, fall apart. In fact, straightforward applications of ERM lead to what has been called "catastrophic forgetting" [McCloskey and Cohen, 1989]. That is, the learner forgets how to perform on past tasks as it is exposed to new tasks.

This paper narrows the gap between ERM and the more human-like description above. In particular, we assume that our learning machine observes, *example by example*, the *continuum of data*

$$(x_1, t_1, y_1), \ldots, (x_i, t_i, y_i), \ldots, (x_n, t_n, y_n) \tag{1}$$

where besides input and target vectors, the learner observes $t_i \in \mathcal{T}$, a *task descriptor* that describes the task associated to the pair (x_i, y_i). Moreover, examples are not drawn iid from a fixed probability distribution over triplets (x, t, y), since a whole sequence of examples from the current task may be observed before switching to the next task. The goal of *learning over a continuum of data* is to construct a model $f : \mathcal{X} \times \mathcal{T}$ able to predict the label associated to a test pair (x, t). In this setting, we face challenges unknown to ERM:

1. *Non-iid input data*: the continuum of data is not *iid* with respect to any fixed probability distribution $P(X, T, Y)$ since, once tasks switch, a whole sequence of examples from the new task may be observed.
2. *Catastrophic forgetting*: learning new tasks may hurt the performance of the learner at previously solved tasks.
3. *Transfer learning*: when the tasks in the continuum are related, there exists an opportunity for transfer learning. This would translate into a faster learning of new tasks, and performance improvements in old tasks.

The rest of this paper is organized as follows. In Section 2, we formalize the problem of learning over a continuum of data, and introduce the metrics necessary to evaluate learners in this scenario. In Section 3, we propose GEM, a model to learn over continuums of data that alleviates forgetting, while transferring beneficial knowledge to past tasks. In Section 4, we compare the performance of GEM to the state-of-the-art. Finally, we conclude by reviewing the related literature in Section 5, and offer some directions for future research in Section 6.

2 A Framework for Continuum Learning

The central object of study in this paper is the *continuum* of data (1), where each triplet (x_i, t_i, y_i) is formed by a feature vector $x_i \in \mathcal{X}_{t_i}$, a task descriptor $t_i \in \mathcal{T}$, and a target vector $y_i \in \mathcal{Y}_{t_i}$. For simplicity, we assume that the continuum is:

1. *Locally iid*: every triplet (x_i, t_i, y_i) satisfies $(x_i, y_i) \overset{iid}{\sim} P(X, Y|t_i)$.
2. *Ordered*: there exists a partial ordering of $\mathcal{T}$, such that $t_i \preceq t_j$ for all $i \leq j$.
3. *Finite*: the continuum contains $n < \infty$ examples, concerning $|\mathcal{T}| = T$ tasks.

While observing (1) *example by example*, our goal is to learn a predictor $f : \mathcal{X} \times \mathcal{T} \to \mathcal{Y}$, which can be queried *at any time* to predict the target vector associated to a test pair (x, t). Such test pair can belong to a task that we have observed in the past, the current task, or a task that we will experience (or not) in the future.

A novel component in (1) is the collection of task descriptors $t_1, \ldots, t_n \in \mathcal{T}$. In the simplest case, the task descriptors are integers, $t_i = i \in \mathbb{Z}$, enumerating the different tasks appearing in the continuum of data. More generally, task descriptors t_i could be structured objects, such as a paragraph of natural language explaining how to solve the i-th task. Rich task descriptors offer an opportunity for zero-shot learning, since the relation between tasks could be inferred using task descriptors alone. Furthermore, task descriptors disambiguate similar learning tasks. In particular, the same input x_i could appear in two different tasks, one requiring classification, and the other requiring regression.

Next, we discuss the training protocol and evaluation metrics to learn over a continuum of data.

2.1 Training Protocol and Evaluation Metrics

Most of the literature about learning over a sequence of tasks [Rusu et al., 2016, Fernando et al., 2017, Kirkpatrick et al., 2017, Rebuffi et al., 2017] describes a setting where i) the number of tasks is small, ii) the number of examples per task is large, iii) the learner performs several epochs over the examples concerning each task, and iv) the only metric reported is the average performance across all tasks. In contrast, we are interested in the "more human-like" setting where i) the number of tasks is large, ii) the number of training examples per task is small, iii) the learner observes the examples concerning each task only once, and iv) we report metrics that measure both transfer and forgetting.

Therefore, at training time we provide the learner with only one example at the time (or a small mini-batch), in the form of a triplet (x_i, t_i, y_i). The learner never experiences the same example twice, and tasks are streamed in sequence.[1]

Besides monitoring its performance across tasks, it is also important to assess the ability of the learner to *transfer* knowledge. More specifically, we would like to measure:

[1] This is without loss in generality, since a future task may *coincide* with a past task.

1. *Backward transfer* (BWT) is the influence that learning a task t has on the performance on a previous task $t' \prec t$. On the one hand, there exists *positive* backward transfer when learning about some task t increases the performance on some preceding task t'. On the other hand, there exists *negative* backward transfer when about some task t decreases the performance on some preceding task t'. Negative backward transfer is also known as *catastrophic forgetting*.
2. *Forward transfer* (FWT) is the influence that learning a task t has on the performance on a future task $t' \succ t$. In particular, *positive* forward transfer is possible when the model is able to perform "zero-shot" learning, perhaps by exploiting the structure available in the task descriptors.

For a principled evaluation, we consider access to a test set for each of the T tasks. After the model finishes learning about the task t_i, we evaluate its *test* performance on all T tasks. By doing so, we construct the matrix $R \in \mathbb{R}^{T \times T}$, where $R_{i,j}$ is the test classification accuracy of the model on task t_j after observing the last sample from task t_i. Letting $\bar{b}$ be the vector of test accuracies for each task at random initialization, we define three metrics:

$$\textbf{Average Accuracy} \ \text{ACC} = \frac{1}{T} \sum_{i=1}^{T} R_{T,i} \tag{2}$$

$$\textbf{Backward Transfer} \ \text{BWT} = \frac{1}{T-1} \sum_{i=1}^{T-1} R_{T,i} - R_{i,i} \tag{3}$$

$$\textbf{Forward Transfer} \ \text{FWT} = \frac{1}{T-1} \sum_{i=2}^{T} R_{i-1,i} - \bar{b}_i. \tag{4}$$

The larger these metrics, the better the model. If two models have similar ACC, the most preferable one is the one with larger BWT and FWT. Note that it is meaningless to discuss backward transfer for the first task, or forward transfer for the last task.

For a fine-grained evaluation that accounts for learning speed, one can build a matrix R with more rows than tasks, by evaluating more often. In the extreme case, the number of rows could equal the number of continuum samples n. Then, the number $R_{i,j}$ is the test accuracy on task t_j after observing the i-th example in the continuum. Plotting each column of R results into a learning curve.

3 Gradient of Episodic Memory (GEM)

In this section, we propose Gradient Episodic Memory (GEM), a framework to learn over a continuum of data, as introduced in Section 2. The main feature of GEM is an *episodic memory* $\mathcal{M}_t$, which stores a subset of the observed examples from task t. For simplicity, we assume integer task descriptors, and use them to index the episodic memory. When using integer task descriptors, one cannot expect significant positive forward transfer (zero-shot learning). Instead, we focus on minimizing negative backward transfer (catastrophic forgetting) by the efficient use of episodic memory.

In practice, the learner has a total budget of M memory locations. If the number of total tasks T is known, we can allocate $m = M/T$ memories for each task. Conversely, if the number of total tasks T is unknown, we can gradually reduce the value of m as we observe new tasks [Rebuffi et al., 2017]. For simplicity, we assume that the memory is populated with the last m examples from each task, although better memory update strategies could be employed (such as building a coreset per task). In the following, we consider predictors f_θ parameterized by $\theta \in \mathbb{R}^p$, and define the loss at the memories from the k-th task as

$$\ell(\theta, \mathcal{M}_k) = \frac{1}{|\mathcal{M}_k|} \sum_{(x_i,k,y_i) \in \mathcal{M}_k} \ell(f_\theta(x_i, k), y_i). \tag{5}$$

Preliminary experiments revealed that minimizing the loss at the current example together with (5) results in overfitting to the examples stored in $\mathcal{M}_k$. As an alternative, we could keep the predictions at past tasks invariant by means of distillation [Rebuffi et al., 2017]. However, this would deem positive backward transfer impossible. Instead, we will use the losses (5) as *inequality constraints*, avoiding their increase but allowing their decrease. In contrast to the state-of-the-art [Kirkpatrick et al., 2017, Rebuffi et al., 2017], our model therefore allows positive backward transfer.

More specifically, when observing the triplet (x, t, y), we solve the following problem:

$$\theta^* = \arg\min_{\theta} \ell(f_\theta(x,t), y) \tag{6}$$
$$\text{such that } \ell(\theta, \mathcal{M}_k) \leq \ell(\theta_{t-1}, \mathcal{M}_k), \text{ for } k \in [1, t-1].$$

In the previous equation, θ_{t-1} are the predictor parameters at the end of learning of task $t-1$.

In the following, we make two key observations to solve (6) efficiently. First, it is unnecessary to store old parameter vectors θ_{t-1}, as long as we guarantee that the loss at previous tasks does not increase after each parameter update. Second, assuming that the function is locally linear (the step of SGD is small) and that the memory is representative of the examples from past tasks, we can diagnose increases in the loss of previous tasks by computing the angle between gradient vectors, as weight updates with positive inner product with $\ell(\theta_{t-1}, \mathcal{M}_k)$ would increase $\ell(\theta_{t-1}, \mathcal{M}_k)$. Then, the constraints in (6) can be rephrased as:

$$\langle g, g_k \rangle := \left\langle \frac{\partial \ell(f_\theta(x,t), y)}{\partial \theta}, \frac{\partial \ell(\theta, \mathcal{M}_k)}{\partial \theta} \right\rangle \geq 0, \text{ for all } k < t. \tag{7}$$

If all of the inequality constraints (7) are satisfied, then the proposed parameter update is unlikely to increase the loss at previous tasks. On the other hand, if one or more of the inequality constraints (7) are violated, then there is at least one previous task that would experience an increase in loss after the parameter update. When violations occur, we project the proposed parameter update to the closest update (in L2 distance) that satisfies all the constraints in (7). Since there is no general closed form solution for computing the projection of a vector onto a cone —the intersection of the half-planes associated to the $(t-1)$ inequality constraints in (7)— we set up a simple inner optimization problem.

This inner optimization problem, which needs to be solved for every parameter update, can be naïvely set up as a problem over p variables (the dimensionality of the parameter vector), a number which runs in the millions for deep neural networks. However, this optimization problem has a very simple geometry, which we can leverage to reduce drastically the number of optimized variables. More specifically, the gradient vector at the current input sample can be decomposed as $g = g^{\|} + g^{\perp}$, where $g^{\|}$ is the component in span$(g_1, ..., g_{t-1})$ and $g^{\perp}$ is the component in the orthogonal complement of such span. Since the number of tasks is much smaller than the size of the parameter vector p, we only need to optimize over the component in span$(g_1, ..., g_{t-1})$. Therefore, state the inner optimization problem as:

1. Decompose g as the sum of $g^{\perp}$ and $g^{\|}$ by solving (problem over $t-1$ variables):
$$\alpha^* = \arg\min_{\alpha} \left\| g - \sum_{i=1}^{t-1} \alpha_i g_i \right\|_2^2, \tag{8}$$
to find: $g^{\|} = \sum_{i=1}^{t-1} \alpha_i^* g_i$, and $g^{\perp} = g - g^{\|}$.
2. Find the optimal component in the span satisfying the constraints (another problem over $t-1$ variables):
$$\beta^* = \arg\min_{\beta} \left\| g^{\|} - \sum_{i=1}^{t-1} \beta_i g_i \right\|_2^2, \text{such that } \langle \sum_{i=1}^{t-1} \beta_i g_i, g_i \rangle \geq 0. \tag{9}$$
This problem can be solved by gradient descent (from a feasible point) until the solution violates a constraint, at which point, the optimization stops and the solution is thresholded.
3. Return the final solution, $g^{\perp} + \bar{g}^{\|}$, with $\bar{g}^{\|} = \sum_{i=1}^{t-1} \beta_i^* g_i$.

Proposition 1. *The proposed inner optimization problem finds the vector closest in L2 norm to g satisfying the constraints in eq.* (7).

Proof. The gradient on the current task decomposes as

$$g = g^{\perp} + g^{\|},$$

where

$$\begin{aligned} g^{\|} &\in S = \text{span}(g_1, ..., g_{t-1}) \\ g^{\perp} &\in S^{\perp} = \text{span}(g_1, ..., g_{t-1})^{\perp}. \end{aligned}$$

In the previous, $g^{\perp}$ is in the orthogonal complement of the span. Next, we find $u^{\|} \in S$ and $u^{\perp} \in S^{\perp}$ such that $u = u^{\|} + u^{\perp}$ is the closest vector to g that satisfies all the constraints in (7). That is,

$$\arg\min_{u} \|g - u\|_2^2 \text{ such that } u \cdot g_k \geq 0, k < t.$$

Expanding the squared norm, we obtain:

$$\begin{aligned} \|g - u\|_2^2 &= \|(g - u^{\perp}) - u^{\|}\|_2^2 \\ & \quad \|g - u^{\perp}\|_2^2 + \|u^{\|}\|_2^2 - 2(g - u^{\perp}) \cdot u^{\|} \\ & \quad \|g - u^{\perp}\|_2^2 + \|u^{\|}\|_2^2 - 2g \cdot u^{\|}. \end{aligned}$$

Furthermore, the constraints can be written as:

$$u \cdot g_k \geq 0 \tag{10}$$

$$(u^{\perp} + u^{\|}) \cdot g_k \geq 0 \tag{11}$$

$$u^{\|} \cdot g_k \geq 0 \tag{12}$$

by leveraging the fact that by definition: $u^{\perp} \cdot u^{\|} = 0$ and $u^{\perp} \cdot g_k = 0$.

Next, we note that the minimization of $u^{\perp}$ and $u^{\|}$ can happen independently, since the two variables do not interact. Trivially, the minimum of $\|g - u^{\perp}\|_2^2$ is achieved when $u^{\perp} = g^{\perp}$. We find this by first computing the component of $g \in S$ (an optimization over $(t-1)$ variables), and then subtracting the original parameter update to find the orthogonal complement, $g^{\perp}$.

The other component, $u^{\|}$, in (10) can instead be found by solving the equivalent problem:

$$\arg\min_{u} \|g - u\|_2^2 \text{ such that } u \cdot g_k \geq 0, k < t, u \in S \tag{13}$$

which again requires an optimization over $(t-1)$ variables, since the solution has to be in S. □

Under the assumption that the memory is populated with representative samples of the distribution, then the loss won't increase neither for the current nor for past tasks. In practice, the model may suffer a small decrease of accuracy over time because the memory is a rough proxy of the full distribution, the loss is a mere surrogate of accuracy, and we may incur some generalization error. Since we consider classification tasks, in our experiments we use the cross-entropy loss with soft targets.

Algorithm 1 Continuum Learning via GEM

procedure TRAINING(f_θ, Continuum) ▷ Training a predictor on a continuum of data
  for t in range(T) **do** ▷ Loop over tasks in order.
    for (x, t, y) in Continuum(t) **do** ▷ Get inputs for each task.
      $(x, y) \rightarrow \mathcal{M}_t$ ▷ Add example to memory. If memory is full, remove oldest example.
      compute g and g_k for $k < t$. ▷ Fprop/Bprop on current minibatch and memories.
      if constraints in (7) are not satisfied **then**
        $g \leftarrow$ PROJECT($g, g_1, \ldots, g_{t-1}$) ▷ Project.
      end if
      $\theta \leftarrow \theta - \eta g$ ▷ Update parameters by one step gradient descent with step size η.
    end for
    return EVALUATE(f_θ, t, Continuum)
  end for
end procedure

procedure EVALUATE(f_θ, t, Continuum) ▷ Evaluate on all tasks from test set.
  for k in range(T) **do** ▷ Evaluate on past, present and future tasks.
    for (x, t_k, y) in Continuum(t_k) **do** ▷ Get inputs for each task.
      eval $f_\theta(x, t_k)$ against y ▷ Compute loss/accuracy.
    end for
    update $R[t, k]$ ▷ Update entries in matrix R
  end for
  return R
end procedure

procedure PROJECT($g, g_1, \ldots, g_{t-1}$) ▷ Find closest vector satisfying all constraints.
  Compute $g^{\perp}$ by solving (8) by gradient descent.
  Compute $\bar{g}^{\|}$ by solving (9) by gradient descent.
  return $g^{\perp} + \bar{g}^{\|}$
end procedure

4 Experiments

In this section, we describe a variety of experiments that assess the performance of GEM to learn over a continuum of data.

4.1 Datasets

We consider the following datasets:

- *MNIST Permutations* [Kirkpatrick et al., 2017], a variant of the MNIST dataset of handwritten digits [LeCun et al., 1998], where each task is transformed by a fixed permutation of pixels. In this dataset, the input distribution for each task is unrelated.
- *MNIST Rotations*, a variant of MNIST where each task has digits rotated by a fixed angle between 0 and 180 degrees.
- *Incremental CIFAR100* [Rebuffi et al., 2017], a variant of the CIFAR object recognition dataset with 100 classes [Krizhevsky, 2009], where each task introduces a new set of classes. For a total number of T tasks, each new task concerns examples from a disjoint subset of $100/T$ classes. Here, the input distribution is similar for all tasks, but different tasks require different output distributions.

For all the datasets, we considered $T = 20$ tasks. On the MNIST datasets, each task has 1000 examples from 10 different classes, while on the CIFAR100 dataset each task has 2500 examples from 5 different classes. The model observes the tasks in sequence, and each example once. The evaluation for each task is performed on the test partition of each dataset.

4.2 Architectures

On the MNIST tasks, we use fully-connected neural networks with two hidden layers of 100 ReLU units. On the CIFAR100 tasks, we use a smaller version of ResNet18 [He et al., 2015], with three times less feature maps across all layers. Furthermore, the network has a final linear classifier per task. This is one simple way to leverage the task descriptor, in order to adapt the output distribution to the subset of classes for each task. We train all the networks and baselines using plain SGD on mini-batches of 10 samples. All hyper-parameters are optimized using a grid-search (see Appendix A.1), and the best results for each model are reported.

4.3 Methods

We compare GEM to several alternatives:

1. *single* predictors trained across all tasks.
2. *independent* predictors per task. The size of each *independent* predictor is T times less the size of *single* predictors. Each new independent predictor can be initialized at random, or be a clone of the last trained predictor (to be decided by grid-search).
3. *multimodal* predictors, with a dedicated input layer per task (only for MNIST datasets).
4. *EWC* [Kirkpatrick et al., 2017], where the loss is regularized to avoid catastrophic forgetting.
5. *iCARL* [Rebuffi et al., 2017], a class-incremental learner that classifies using a nearest-exemplar algorithm, and prevents catastrophic forgetting by using an episodic memory. iCARL requires the same input representation across tasks, so this method only applies to our experiment on CIFAR100.

4.4 Results

Figure 1 (left) summarizes the average accuracy (ACC, Equation 2), backward transfer (BWT, Equation 3) and forward transfer (FWT, Equation 4) for all datasets and methods. We provide the full evaluation matrices R in Appendix B. Overall, GEM performs similarly or better than the multimodal model (which is very well suited to the MNIST tasks). GEM minimizes backward transfer, while exhibiting negligible or positive forward transfer.

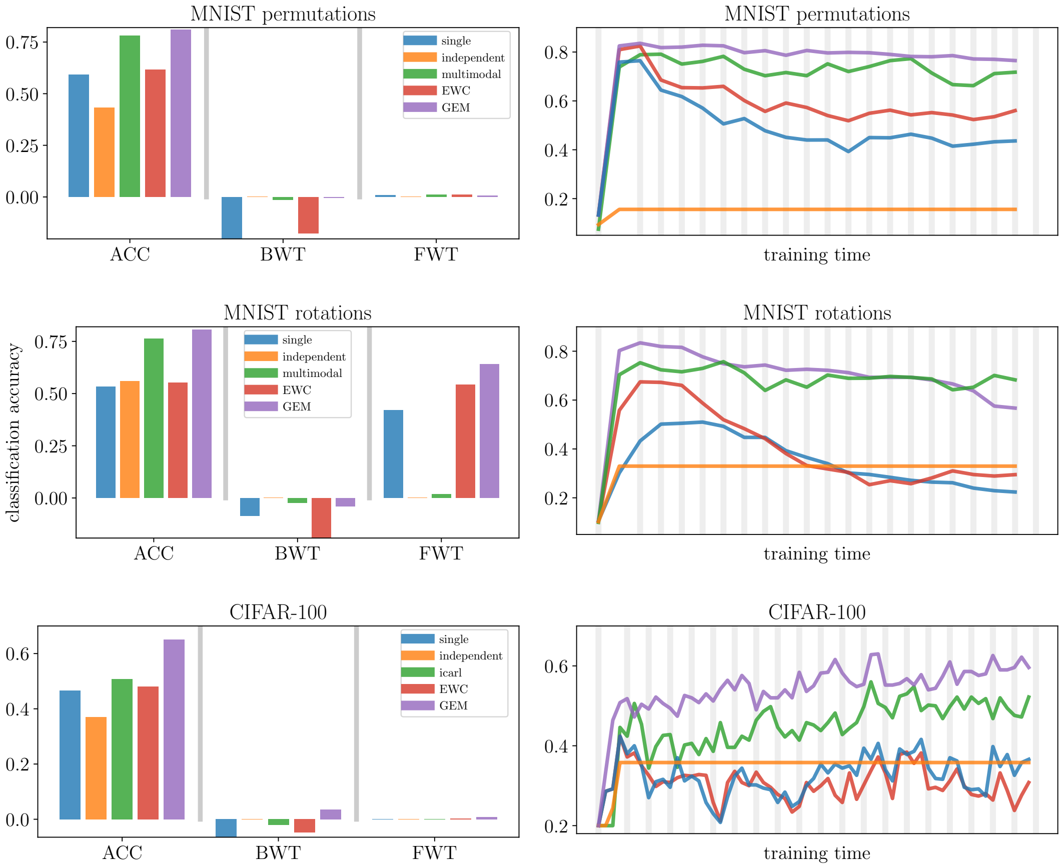

Figure 1: Left: ACC, BWT, and FWT for all datasets and methods. Right: evolution of the test accuracy at the first task, as more tasks (denoted by vertical gray bars) are introduced.

number of memories	200	1,280	2,560	5,120
GEM	0.487	0.579	0.633	0.654
iCARL	0.436	0.494	0.500	0.508

Table 1: ACC as a function of the episodic memory size for GEM and iCARL on CIFAR100.

Figure 1 (right) shows the evolution of the test accuracy of the first task throughout the continuum of data. GEM exhibits minimal forgetting, and positive backward transfer in the experiment on CIFAR100. Finally, Table 1 shows the final performance in the CIFAR-100 experiment for GEM and iCARL as a function of the size of their episodic memory. As seen in the table, the final accuracy of GEM is an increasing function of the size of the episodic memory, eliminating the need to tune carefully this hyper-parameter.

5 Related work

Continuum learning is one instance of *Lifelong learning* [Thrun and Pratt, 2012, Thrun, 1998, 1996], which enjoys several implementations [Carlson et al., 2010, Ruvolo and Eaton, 2013, Ring, 1997]. In this work, we abstract from the specifics of the particular application, in order to provide general evaluation metrics and learning algorithms. While there has been work towards theoretical understanding of lifelong learning [Baxter, 2000, Balcan et al., 2015, Pentina and Urner, 2016], it has been restricted to *linear* models. The *CommAI project* [Mikolov et al., 2015, Baroni et al., 2017] shares our same motivations, but focuses on highly structured task descriptors. In contrast, we focus on the problem of catastrophic forgetting [McCloskey and Cohen, 1989, French, 1999, Ratcliff, 1990, McClelland et al., 1995, Goodfellow et al., 2013]. Several approaches have been proposed to avoid catastrophic forgetting. The simplest approach in neural networks is to freeze early layers, while cloning and fine-tuning later layers on the new task [Oquab et al., 2014]. This relates to methods that leverage a modular structure of the network with primitives that can be shared across tasks [Rusu et al., 2016, Fernando et al., 2017, Aljundi et al., 2016, Denoyer and Gallinari, 2015, Eigen et al., 2014]. Unfortunately, it has been very hard to scale up these methods to lots of modules and tasks, given the combinatorial number of combinations of modules.

Our approach is most similar to the regularization approaches that consider a single model, but modify its learning objective to prevent catastrophic forgetting. Within this class of methods, there are approaches that leverage "synaptic" memory [Kirkpatrick et al., 2017, Zenke et al., 2017], whose learning rates are adjusted to minimize changes in parameters important for previous tasks. Other approaches are instead based on "episodic" memory [Jung et al., 2016, Li and Hoiem, 2016, Rannen Triki et al., 2017, Rebuffi et al., 2017], where examples from previous tasks are stored and replayed to maintain predictions invariant by means of distillation [Hinton et al., 2015]. GEM is related to these latter approaches but, unlike them, allows positive backward transfer.

More generally, there are a variety of setups in the machine learning literature related to continuum learning. *Multitask learning* [Caruana, 1998] considers the problem of maximizing the performance of a learning machine across a variety of tasks, but the setup assumes simultaneous access to all the tasks at once. Similarly, *transfer learning* [Pan and Yang, 2010] and *domain adaptation* [Ben-David et al., 2010] assume the simultaneous availability of multiple learning tasks, but focus at improving the performance at one of them in particular. *Zero-shot learning* [Lampert et al., 2009, Palatucci et al., 2009] and *one-shot learning* [Fei-Fei et al., 2003, Vinyals et al., 2016, Santoro et al., 2016, Bertinetto et al., 2016] aim at performing well on unseen tasks, but ignore the catastrophic forgetting of previously learned tasks. *Curriculum learning* considers learning a sequence of data [Bengio et al., 2009], or a sequence of tasks [Pentina et al., 2015], sorted by increasing difficulty.

6 Conclusion

We formalized the scenario of *learning over a continuum of data*. First, we defined training and evaluation protocols to assess the quality of models in terms of their *accuracy*, as well as their ability to transfer knowledge *forward* and *backward* between tasks. Second, we introduced GEM, a simple model that leverages an episodic memory to avoid forgetting and favor positive backward transfer. Our experiments demonstrate the competitive performance of GEM against the state-of-the-art.

In its current form, our model has two major limitations. First, GEM does not leverage structured task descriptors, which may be exploited to obtain positive forward transfer. Second, we did not investigate advanced ways to learn how to populate the memory (such as building *coresets of tasks* [Lucic et al., 2017]), which is key to prevent forgetting more efficiently.

Acknowledgements

We are grateful to M. Baroni, L. Bottou, M. Nickel, Y. Olivier and A. Szlam for their suggestions and insights.

References

R. Aljundi, P. Chakravarty, and T. Tuytelaars. Expert gate: Lifelong learning with a network of experts. *CVPR*, 2016.

M.-F. Balcan, A. Blum, and S. Vempola. Efficient representations for lifelong learning and autoencoding. *COLT*, 2015.

M. Baroni, A. Joulin, A. Jabri, G. Kruszewski, A. Lazaridou, K. Simonic, and T. Mikolov. CommAI: Evaluating the first steps towards a useful general AI. *arXiv*, 2017.

J. Baxter. A model of inductive bias learning. *JAIR*, 2000.

S. Ben-David, J. Blitzer, K. Crammer, A. Kulesza, F. Pereira, and J. Wortman Vaughan. A theory of learning from different domains. *Machine Learning Journal*, 2010.

Y. Bengio, J. Louradour, R. Collobert, and J. Weston. Curriculum learning. *ICML*, 2009.

L. Bertinetto, J. Henriques, J. Valmadre, P. Torr, and A. Vedaldi. Learning feed-forward one-shot learners. *NIPS*, 2016.

A. Carlson, J. Betteridge, B. Kisiel, B. Settles, E. R. Hruschka, and T. M. Mitchell. Toward an architecture for never-ending language learning. *AAAI*, 2010.

R. Caruana. Multitask learning. In *Learning to learn*. Springer, 1998.

L. Denoyer and P. Gallinari. Deep sequential neural networks. *EWRL*, 2015.

D. Eigen, I. Sutskever, and M. Ranzato. Learning factored representations in a deep mixture of experts. *ICLR*, 2014.

L. Fei-Fei, R. Fergus, and P. Perona. A Bayesian approach to unsupervised one-shot learning of object categories. *ICCV*, 2003.

C. Fernando, D. Banarse, C. Blundell, Y. Zwols, D. Ha, A. A. Rusu, A. Pritzel, and D. Wierstra. PathNet: Evolution channels gradient descent in super neural networks. *arXiv*, 2017.

R. M. French. Catastrophic forgetting in connectionist networks. *Trends in cognitive sciences*, 1999.

I. J. Goodfellow, M. Mirza, D. Xiao, A. Courville, and Y. Bengio. An Empirical Investigation of Catastrophic Forgetting in Gradient-Based Neural Networks. *arXiv*, 2013.

K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. *arXiv*, 2015.

G. Hinton, O. Vinyals, and J. Dean. Distilling the knowledge in a neural network. *arXiv*, 2015.

H. Jung, J. Ju, M. Jung, and J. Kim. Less-forgetting Learning in Deep Neural Networks. *arXiv*, 2016.

J. Kirkpatrick, R. Pascanu, N. Rabinowitz, J. Veness, G. Desjardins, A. A. Rusu, K. Milan, J. Quan, T. Ramalho, A. Grabska-Barwinska, et al. Overcoming catastrophic forgetting in neural networks. *PNAS*, 2017.

A. Krizhevsky. Learning multiple layers of features from tiny images. Technical report, Technical report, University of Toronto, 2009.

C. Lampert, H. Nickisch, and S. Harmeling. Learning to detect unseen object classes by between-class attribute transfer. *CVPR*, 2009.

Y. LeCun, C. Cortes, and C. J. Burges. The MNIST database of handwritten digits, 1998. URL `http://yann.lecun.com/exdb/mnist/`.

Z. Li and D. Hoiem. Learning without forgetting. *ECCV*, 2016.

M. Lucic, M. Faulkner , A. Krause, and D. Feldman. Training Mixture Models at Scale via Coresets. *arXiv*, 2017.

J. L. McClelland, B. L. McNaughton, and R. C. O'reilly. Why there are complementary learning systems in the hippocampus and neocortex: insights from the successes and failures of connectionist models of learning and memory. *Psychological review*, 1995.

M. McCloskey and N. J. Cohen. Catastrophic interference in connectionist networks: The sequential learning problem. *Psychology of learning and motivation*, 1989.

T. Mikolov, A. Joulin, and M. Baroni. A roadmap towards machine intelligence. *arXiv*, 2015.

M. Oquab, L. Bottou, I. Laptev, and J. Sivic. Learning and transferring mid-level image representations using convolutional neural networks. *CVPR*, 2014.

M. Palatucci, D. A. Pomerleau, G. E. Hinton, and T. Mitchell. Zero-shot learning with semantic output codes. *NIPS*, 2009.

S. J. Pan and Q. Yang. A survey on transfer learning. *TKDE*, 2010.

A. Pentina and R. Urner. Lifelong learning with weighted majority votes. *NIPS*, 2016.

A. Pentina, V. Sharmanska, and C. H. Lampert. Curriculum learning of multiple tasks. *CVPR*, 2015.

A. Rannen Triki, R. Aljundi, M. B. Blaschko, and T. Tuytelaars. Encoder Based Lifelong Learning. *arXiv*, 2017.

R. Ratcliff. Connectionist models of recognition memory: Constraints imposed by learning and forgetting functions. *Psychological review*, 1990.

S.-A. Rebuffi, A. Kolesnikov, G. Sperl, and C. H. Lampert. iCaRL: Incremental classifier and representation learning. *CVPR*, 2017.

M. B. Ring. CHILD: A first step towards continual learning. *Machine Learning*, 1997.

A. A. Rusu, N. C. Rabinowitz, G. Desjardins, H. Soyer, J. Kirkpatrick, K. Kavukcuoglu, R. Pascanu, and R. Hadsell. Progressive neural networks. *NIPS*, 2016.

P. Ruvolo and E. Eaton. ELLA: An Efficient Lifelong Learning Algorithm. *ICML*, 2013.

A. Santoro, S. Bartunov, M. Botvinick, D. Wierstra, and T. Lillicrap. One-shot learning with memory-augmented neural networks. *arXiv*, 2016.

S. Thrun. Is learning the n-th thing any easier than learning the first? *NIPS*, 1996.

S. Thrun. Lifelong learning algorithms. In *Learning to learn*. Springer, 1998.

S. Thrun and L. Pratt. *Learning to learn*. Springer Science & Business Media, 2012.

V. Vapnik. *Statistical learning theory*. Wiley New York, 1998.

O. Vinyals, C. Blundell, T. Lillicrap, and D. Wierstra. Matching networks for one shot learning. *NIPS*, 2016.

F. Zenke, B. Poole, and S. Ganguli. Improved multitask learning through synaptic intelligence. *arXiv*, 2017.

A Supplementary Material

A.1 Hyper-parameter Selection

The hyper-paramter values used during grid search and the best values found are reported below.

```
Single Predictor:
learning rate: [1.0, 0.3, 0.1, 0.03, 0.01, 0.003, 0.001]

Independent Predictors:
learning rate: [1.0, 0.3, 0.1, 0.03, 0.01, 0.003, 0.001]
finetune: [no, yes]

Multi-Modal Predictor:
learning rate: [1.0, 0.3, 0.1, 0.03, 0.01, 0.003, 0.001]

EWC:
learning rate: [1.0, 0.3, 0.1, 0.03, 0.01, 0.003, 0.001]

regularization: [1, 3, 10, 30, 100, 300, 1000, 3000, 10000, 30000]

iCARL:
learning rate: [10.0, 3.0, 1.0, 0.3, 0.1, 0.03, 0.01, 0.003, 0.001]
regularization: [1, 3, 10, 30, 100, 300, 1000, 3000, 10000, 30000] # MNIST
                [0.1, 0.3, 1, 3, 10, 30] # CIFAR
memory_size:    [100, 1000] # for MNIST
                [200, 1280, 2560, 5120] # for CIFAR

GEM:
learning rate: [1.0, 0.3, 0.1, 0.03]
inner learning rate: [0.1, 1.0, 10] # MNIST
inner learning rate: [1.0, 3.0, 10.0] # CIFAR
soft_targets: [0.25, 0.5, 0.75, 1]
memory_size: [200, 1280, 5120]
```

The best values we found are:

```
MNIST PERMUTATION
Single: learning rate=0.03
Independent: learning rate=0.03, finetune=yes
Multi-Modal: learning rate=0.1
EWC: learning rate=0.1, regularization=3
GEM: learning rate=0.3, memory size=5120, target=0.5, inner learning rate=1.0

MNIST ROTATION
Single: learning rate=0.003
Independent: learning rate=0.1, finetune=yes
Multi-Modal: learning rate=0.1
EWC: learning rate=0.01, regularization=1000
GEM: learning rate=0.3, memory size=5120, target=0.25, inner learning rate=0.1

CIFAR100
Single: learning rate=1.0
Independent: learning rate=0.3, finetune=yes
iCARL: learning rate=0.3, regularization=1.0, memory size=5120
EWC: learning rate=1.0, regularization=1.0
GEM: learning rate=0.1, memory size=5120, target=0.75, inner learning rate=3.0
```

B Full experiments

In this section we report the evaluation matrices R for each model and dataset. The first row of each matrix (above the line) is the baseline accuracy $\bar{b}$, accuracy before any training takes place. The entry (i, j) of the matrix R is the accuracy of the j-th task just after training the i-th task.

B.1 MNIST permutations

B.1.1 Model *single*

0.1330	0.1199	0.1070	0.0825	0.0609	0.0832	0.1385	0.1123	0.0736	0.1190	0.0666	0.0890	0.0885	0.0723	0.1083	0.0524	0.0976	0.0871	0.1143	0.0743
0.7588	0.1102	0.0685	0.1034	0.0553	0.1091	0.0893	0.1057	0.0710	0.0895	0.0634	0.1009	0.0862	0.0837	0.0926	0.0883	0.1176	0.0790	0.1707	0.0760
0.7645	0.7919	0.1014	0.1507	0.0822	0.0787	0.1177	0.0659	0.0948	0.0807	0.0601	0.0799	0.0973	0.0646	0.0810	0.0898	0.1244	0.1269	0.1564	0.0642
0.6443	0.7201	0.7454	0.1624	0.0626	0.0615	0.1469	0.0566	0.1121	0.0799	0.0691	0.0883	0.0724	0.0819	0.0898	0.0795	0.1541	0.1087	0.1575	0.0691
0.6177	0.7113	0.7739	0.8092	0.0579	0.0733	0.1122	0.0726	0.1110	0.0879	0.0783	0.0578	0.0776	0.0747	0.0944	0.0851	0.1379	0.1063	0.1386	0.0848
0.5706	0.6799	0.7742	0.7683	0.7987	0.0882	0.0865	0.0655	0.1054	0.1001	0.0751	0.0626	0.0850	0.0745	0.0877	0.0607	0.1286	0.1086	0.1360	0.0924
0.5059	0.7345	0.7694	0.7826	0.7976	0.8066	0.1042	0.0648	0.0897	0.0616	0.0420	0.0690	0.0883	0.0525	0.0960	0.0758	0.1309	0.0770	0.1545	0.0620
0.5277	0.6399	0.7078	0.7363	0.7458	0.7607	0.7863	0.0965	0.0971	0.1046	0.0649	0.0852	0.0519	0.0937	0.1039	0.0715	0.1463	0.0915	0.1256	0.0813
0.4778	0.6454	0.7278	0.6671	0.7650	0.7522	0.7933	0.7968	0.1026	0.0853	0.0529	0.0940	0.0715	0.0656	0.1065	0.0545	0.1149	0.1062	0.1289	0.0627
0.4509	0.6057	0.6996	0.6797	0.6808	0.7339	0.7836	0.7518	0.7843	0.0784	0.0683	0.0834	0.0526	0.0932	0.1082	0.0595	0.1017	0.0981	0.1286	0.0730
0.4399	0.5193	0.6484	0.5982	0.5708	0.7108	0.7506	0.6755	0.7827	0.7999	0.0678	0.0944	0.0536	0.1036	0.1135	0.0547	0.1320	0.0890	0.1736	0.0929
0.4402	0.5285	0.6595	0.6486	0.6649	0.7004	0.7520	0.7219	0.7619	0.7737	0.8028	0.1009	0.0523	0.1058	0.1233	0.0463	0.1230	0.1060	0.1925	0.0646
0.3926	0.4295	0.5441	0.4534	0.5593	0.5891	0.6685	0.6953	0.7340	0.7030	0.7979	0.7971	0.0542	0.1038	0.1267	0.0580	0.1264	0.1063	0.1195	0.0811
0.4499	0.4472	0.5681	0.5484	0.6013	0.5574	0.7042	0.7168	0.7018	0.7261	0.7859	0.7812	0.7999	0.1155	0.1269	0.0781	0.1094	0.1231	0.1794	0.0930
0.4490	0.4532	0.6656	0.5346	0.5604	0.6074	0.6723	0.6642	0.6784	0.7073	0.7689	0.7656	0.7431	0.8154	0.1233	0.0721	0.1144	0.1162	0.2066	0.0683
0.4637	0.4158	0.6543	0.5917	0.5833	0.5585	0.7061	0.5965	0.7032	0.6926	0.7589	0.7539	0.6877	0.7721	0.8194	0.0600	0.1142	0.1060	0.2245	0.0923
0.4475	0.3708	0.6201	0.4449	0.4766	0.5386	0.6616	0.4843	0.6801	0.6688	0.6724	0.7322	0.7913	0.7734	0.7316	0.7711	0.1109	0.1029	0.1997	0.0958
0.4147	0.3247	0.5326	0.3737	0.4312	0.4932	0.5572	0.4299	0.6532	0.6313	0.6039	0.7091	0.7125	0.6957	0.7421	0.7140	0.7848	0.1203	0.1734	0.1009
0.4227	0.3550	0.5108	0.3773	0.4545	0.5300	0.5731	0.4664	0.6434	0.5878	0.6342	0.6680	0.7089	0.6892	0.7133	0.6983	0.7840	0.8144	0.1739	0.1113
0.4326	0.3389	0.5207	0.4197	0.4776	0.5351	0.5488	0.4440	0.6421	0.5661	0.6362	0.6205	0.6836	0.7003	0.7438	0.7593	0.7680	0.7428	0.8150	0.1016
0.4362	0.3217	0.5002	0.4027	0.4553	0.4714	0.5825	0.4424	0.5904	0.5426	0.6494	0.5864	0.7006	0.6598	0.6825	0.7097	0.7720	0.7423	0.7960	0.8125

```
Final Accuracy: 0.5928
Backward: -0.2027
Forward:  0.0091
```

B.1.2 Model *independent*

0.0936	0.0995	0.0884	0.0893	0.0784	0.1000	0.1108	0.0965	0.1243	0.1048	0.0819	0.1115	0.0999	0.0762	0.1060	0.1260	0.0930	0.1075	0.1126	0.1092
0.1560	0.0995	0.0884	0.0893	0.0784	0.1000	0.1108	0.0965	0.1243	0.1048	0.0819	0.1115	0.0999	0.0762	0.1060	0.1260	0.0930	0.1075	0.1126	0.1092
0.1560	0.2346	0.0884	0.0893	0.0784	0.1000	0.1108	0.0965	0.1243	0.1048	0.0819	0.1115	0.0999	0.0762	0.1060	0.1260	0.0930	0.1075	0.1126	0.1092
0.1560	0.2346	0.3733	0.0893	0.0784	0.1000	0.1108	0.0965	0.1243	0.1048	0.0819	0.1115	0.0999	0.0762	0.1060	0.1260	0.0930	0.1075	0.1126	0.1092
0.1560	0.2346	0.3733	0.3043	0.0784	0.1000	0.1108	0.0965	0.1243	0.1048	0.0819	0.1115	0.0999	0.0762	0.1060	0.1260	0.0930	0.1075	0.1126	0.1092
0.1560	0.2346	0.3733	0.3043	0.4104	0.1000	0.1108	0.0965	0.1243	0.1048	0.0819	0.1115	0.0999	0.0762	0.1060	0.1260	0.0930	0.1075	0.1126	0.1092
0.1560	0.2346	0.3733	0.3043	0.4104	0.3086	0.1108	0.0965	0.1243	0.1048	0.0819	0.1115	0.0999	0.0762	0.1060	0.1260	0.0930	0.1075	0.1126	0.1092
0.1560	0.2346	0.3733	0.3043	0.4104	0.3086	0.4250	0.0965	0.1243	0.1048	0.0819	0.1115	0.0999	0.0762	0.1060	0.1260	0.0930	0.1075	0.1126	0.1092
0.1560	0.2346	0.3733	0.3043	0.4104	0.3086	0.4250	0.4087	0.1243	0.1048	0.0819	0.1115	0.0999	0.0762	0.1060	0.1260	0.0930	0.1075	0.1126	0.1092
0.1560	0.2346	0.3733	0.3043	0.4104	0.3086	0.4250	0.4087	0.4998	0.1048	0.0819	0.1115	0.0999	0.0762	0.1060	0.1260	0.0930	0.1075	0.1126	0.1092
0.1560	0.2346	0.3733	0.3043	0.4104	0.3086	0.4250	0.4087	0.4998	0.4891	0.0819	0.1115	0.0999	0.0762	0.1060	0.1260	0.0930	0.1075	0.1126	0.1092
0.1560	0.2346	0.3733	0.3043	0.4104	0.3086	0.4250	0.4087	0.4998	0.4891	0.3986	0.1115	0.0999	0.0762	0.1060	0.1260	0.0930	0.1075	0.1126	0.1092
0.1560	0.2346	0.3733	0.3043	0.4104	0.3086	0.4250	0.4087	0.4998	0.4891	0.3986	0.4313	0.0999	0.0762	0.1060	0.1260	0.0930	0.1075	0.1126	0.1092
0.1560	0.2346	0.3733	0.3043	0.4104	0.3086	0.4250	0.4087	0.4998	0.4891	0.3986	0.4313	0.5208	0.0762	0.1060	0.1260	0.0930	0.1075	0.1126	0.1092
0.1560	0.2346	0.3733	0.3043	0.4104	0.3086	0.4250	0.4087	0.4998	0.4891	0.3986	0.4313	0.5208	0.5420	0.1060	0.1260	0.0930	0.1075	0.1126	0.1092
0.1560	0.2346	0.3733	0.3043	0.4104	0.3086	0.4250	0.4087	0.4998	0.4891	0.3986	0.4313	0.5208	0.5420	0.5549	0.1260	0.0930	0.1075	0.1126	0.1092
0.1560	0.2346	0.3733	0.3043	0.4104	0.3086	0.4250	0.4087	0.4998	0.4891	0.3986	0.4313	0.5208	0.5420	0.5549	0.4967	0.0930	0.1075	0.1126	0.1092
0.1560	0.2346	0.3733	0.3043	0.4104	0.3086	0.4250	0.4087	0.4998	0.4891	0.3986	0.4313	0.5208	0.5420	0.5549	0.4967	0.4945	0.1075	0.1126	0.1092
0.1560	0.2346	0.3733	0.3043	0.4104	0.3086	0.4250	0.4087	0.4998	0.4891	0.3986	0.4313	0.5208	0.5420	0.5549	0.4967	0.4945	0.5115	0.1126	0.1092
0.1560	0.2346	0.3733	0.3043	0.4104	0.3086	0.4250	0.4087	0.4998	0.4891	0.3986	0.4313	0.5208	0.5420	0.5549	0.4967	0.4945	0.5115	0.5316	0.1092
0.1560	0.2346	0.3733	0.3043	0.4104	0.3086	0.4250	0.4087	0.4998	0.4891	0.3986	0.4313	0.5208	0.5420	0.5549	0.4967	0.4945	0.5115	0.5316	0.5352

```
Final Accuracy: 0.4313
Backward: 0.0000
Forward:  0.0000
```

B.1.3 Model *multitask*

0.0749	0.1152	0.0601	0.0885	0.0826	0.0856	0.0925	0.0703	0.1079	0.0891	0.1029	0.1092	0.0866	0.1014	0.1575	0.1005	0.1083	0.1038	0.0857	0.0759
0.7386	0.1476	0.1319	0.1498	0.1056	0.1163	0.1409	0.1176	0.1171	0.1245	0.1080	0.1388	0.1037	0.0676	0.1379	0.1186	0.1402	0.1116	0.0806	0.1738
0.7890	0.8267	0.1502	0.1429	0.0992	0.1164	0.1537	0.1167	0.1147	0.1463	0.0868	0.1547	0.0909	0.0865	0.1443	0.1230	0.0881	0.0845	0.0597	0.1547
0.7912	0.8317	0.8038	0.1313	0.0896	0.1147	0.1459	0.1076	0.1244	0.1444	0.0775	0.1470	0.0825	0.1034	0.1048	0.1165	0.1233	0.0841	0.0781	0.1401
0.7507	0.8310	0.7877	0.8275	0.0902	0.1129	0.1457	0.1014	0.1273	0.1522	0.0873	0.1443	0.0887	0.1178	0.1151	0.1268	0.1315	0.0832	0.0741	0.1856
0.7617	0.8101	0.8122	0.8298	0.8047	0.1117	0.1558	0.0845	0.1369	0.1466	0.0817	0.1568	0.0992	0.1112	0.1170	0.1063	0.0743	0.0829	0.0683	0.1372
0.7824	0.8264	0.8244	0.8422	0.8060	0.7997	0.1272	0.0758	0.1371	0.1439	0.0606	0.1309	0.0907	0.0909	0.1273	0.1284	0.1281	0.0837	0.0722	0.1592
0.7295	0.8229	0.7500	0.8389	0.8416	0.7561	0.6986	0.0737	0.1328	0.1495	0.0558	0.1245	0.0773	0.0876	0.1268	0.1201	0.1166	0.0633	0.0687	0.1383
0.7030	0.8146	0.7469	0.8214	0.8418	0.7448	0.7127	0.8105	0.1517	0.1429	0.0821	0.1243	0.0787	0.0741	0.1427	0.1275	0.1325	0.0636	0.0714	0.1441
0.7161	0.8055	0.8154	0.7967	0.8007	0.8277	0.6983	0.7722	0.7396	0.1320	0.0815	0.1553	0.0808	0.0706	0.1211	0.1091	0.1063	0.0674	0.0801	0.1217
0.7032	0.7990	0.7871	0.8045	0.8266	0.7874	0.6547	0.8092	0.7754	0.7886	0.0818	0.1088	0.0711	0.0657	0.1163	0.1013	0.0998	0.0718	0.0664	0.1139
0.7517	0.8199	0.8089	0.8148	0.8266	0.7637	0.6925	0.8313	0.7436	0.8032	0.8144	0.1081	0.0688	0.0626	0.1187	0.0974	0.0918	0.0662	0.0594	0.1197
0.7201	0.8016	0.7813	0.8095	0.8250	0.7571	0.6529	0.8368	0.7341	0.7546	0.8037	0.7942	0.0659	0.0696	0.1209	0.0988	0.0906	0.0705	0.0570	0.1135
0.7405	0.7941	0.7916	0.7896	0.7806	0.7766	0.6998	0.8108	0.6968	0.7627	0.7337	0.7726	0.7711	0.0705	0.1384	0.0973	0.1034	0.0917	0.0748	0.1508
0.7648	0.8266	0.7729	0.8094	0.7958	0.7259	0.7081	0.8028	0.6923	0.7925	0.7489	0.7475	0.8131	0.8124	0.1301	0.1006	0.1128	0.0897	0.0952	0.1505
0.7730	0.8384	0.7747	0.8203	0.8086	0.7299	0.7306	0.8131	0.6985	0.7798	0.7468	0.7475	0.8082	0.8135	0.8499	0.1092	0.1031	0.0748	0.0909	0.1263
0.7141	0.8312	0.7480	0.8123	0.8186	0.7118	0.7003	0.8026	0.7389	0.7806	0.7603	0.7499	0.8258	0.8295	0.8200	0.7997	0.0951	0.0732	0.0777	0.1436
0.6667	0.7936	0.7209	0.8093	0.8369	0.7275	0.6810	0.8101	0.7810	0.7476	0.7738	0.7657	0.8146	0.8450	0.7854	0.7932	0.7281	0.0761	0.0817	0.1255
0.6624	0.7721	0.6933	0.7855	0.8037	0.6900	0.6450	0.7638	0.7618	0.7330	0.7913	0.7406	0.8094	0.8045	0.7672	0.7472	0.7215	0.8191	0.0806	0.1109
0.7118	0.8224	0.7611	0.8099	0.8095	0.7389	0.7102	0.7796	0.7422	0.8061	0.7477	0.7626	0.8269	0.8162	0.8061	0.7754	0.7231	0.8053	0.8346	0.1251
0.7175	0.8237	0.7653	0.8234	0.8032	0.7436	0.6887	0.7942	0.7332	0.8000	0.7599	0.7607	0.8155	0.8309	0.8062	0.7798	0.6930	0.8086	0.8346	0.8251

```
Final Accuracy: 0.7804
Backward: -0.0140
Forward:  0.0117
```

B.1.4 Model *EWC*

0.1330	0.1199	0.1070	0.0825	0.0609	0.0832	0.1385	0.1123	0.0736	0.1190	0.0666	0.0890	0.0885	0.0723	0.1083	0.0524	0.0976	0.0871	0.1143	0.0743
0.8097	0.1036	0.0716	0.1110	0.0582	0.0944	0.1032	0.1344	0.0693	0.0648	0.0718	0.1016	0.0834	0.0876	0.1021	0.0963	0.1146	0.0899	0.1833	0.0953
0.8248	0.8600	0.0817	0.1337	0.0703	0.0887	0.1475	0.1074	0.1012	0.0657	0.0702	0.0834	0.0950	0.0464	0.0887	0.0848	0.1412	0.1419	0.1630	0.0849
0.6855	0.8249	0.8232	0.1484	0.0658	0.1090	0.1172	0.0749	0.1098	0.0892	0.0664	0.0514	0.0664	0.0490	0.1101	0.1028	0.1760	0.0809	0.1758	0.0727
0.6543	0.7579	0.7906	0.8021	0.0645	0.0889	0.0865	0.0469	0.1199	0.0655	0.0745	0.0798	0.0732	0.0898	0.0977	0.1049	0.1386	0.0864	0.1194	0.0806
0.6531	0.7335	0.7976	0.7631	0.7956	0.1148	0.0682	0.0643	0.1302	0.0702	0.0721	0.0567	0.0872	0.0757	0.1089	0.0861	0.1478	0.1022	0.1124	0.0992
0.6596	0.7511	0.7744	0.7767	0.7829	0.7903	0.0985	0.0691	0.0986	0.0883	0.0691	0.0589	0.0687	0.0559	0.1142	0.1127	0.1548	0.0998	0.1447	0.0729
0.6015	0.7008	0.7175	0.7362	0.7417	0.7015	0.7471	0.0837	0.0842	0.0911	0.0499	0.1095	0.0330	0.0977	0.1220	0.1062	0.1443	0.0969	0.0950	0.0850
0.5570	0.7306	0.7560	0.6819	0.7471	0.6824	0.7901	0.7914	0.1175	0.1094	0.0650	0.0937	0.0556	0.0795	0.1043	0.0695	0.1216	0.1027	0.1121	0.0873
0.5914	0.6679	0.7668	0.6836	0.7224	0.7101	0.7591	0.6855	0.6403	0.0764	0.0787	0.1011	0.0514	0.0975	0.1324	0.0714	0.1062	0.0903	0.1117	0.0850
0.5729	0.6278	0.7034	0.7276	0.6947	0.7016	0.7183	0.7169	0.7645	0.7981	0.0688	0.1233	0.0811	0.1079	0.1210	0.0801	0.1585	0.0898	0.1091	0.1056
0.5401	0.5669	0.7135	0.6355	0.6847	0.5941	0.7320	0.6857	0.7187	0.7578	0.7740	0.1190	0.0874	0.1140	0.1302	0.0696	0.1328	0.1202	0.1145	0.0675
0.5188	0.5318	0.6382	0.5364	0.6547	0.5606	0.6240	0.6701	0.7090	0.7001	0.7827	0.7971	0.0636	0.1070	0.1350	0.0699	0.1339	0.1250	0.0865	0.0946
0.5493	0.5092	0.6698	0.6100	0.7084	0.5745	0.6552	0.6517	0.6590	0.7125	0.8000	0.7721	0.8051	0.1295	0.1321	0.0955	0.1169	0.1083	0.1107	0.0862
0.5621	0.5415	0.7042	0.5342	0.6561	0.5416	0.7030	0.6006	0.5634	0.6602	0.7157	0.7415	0.7530	0.7916	0.1493	0.1005	0.1198	0.1075	0.1282	0.0763
0.5428	0.5385	0.6986	0.6480	0.6899	0.4990	0.7007	0.5850	0.6323	0.7183	0.7068	0.7538	0.7508	0.7980	0.8444	0.0725	0.1073	0.1123	0.1344	0.0900
0.5521	0.5259	0.6907	0.5502	0.6146	0.5481	0.6738	0.5513	0.7044	0.7014	0.6681	0.7323	0.7499	0.7971	0.7602	0.7969	0.1062	0.1112	0.1247	0.0994
0.5422	0.4118	0.6019	0.4974	0.6052	0.5174	0.5981	0.4574	0.6696	0.5687	0.6087	0.7060	0.6631	0.6359	0.6942	0.7119	0.6992	0.1501	0.1445	0.1116
0.5236	0.5038	0.6439	0.5309	0.6249	0.5420	0.6497	0.5540	0.6824	0.5812	0.6201	0.6993	0.6717	0.6577	0.6338	0.6762	0.7080	0.8311	0.1217	0.1074
0.5353	0.4823	0.6746	0.5249	0.6032	0.5848	0.6267	0.5078	0.6284	0.5640	0.5784	0.6382	0.6958	0.6056	0.7189	0.6007	0.6836	0.7893	0.8224	0.1102
0.5603	0.4639	0.6427	0.4936	0.5961	0.5801	0.5953	0.5040	0.6441	0.4516	0.6180	0.5954	0.6835	0.5660	0.6898	0.6748	0.6697	0.7033	0.7627	0.8007

```
Final Accuracy: 0.6148
Backward: -0.1762
Forward:  0.0116
```

B.1.5 Model *GEM*

0.1330	0.1199	0.1070	0.0825	0.0609	0.0832	0.1385	0.1123	0.0736	0.1190	0.0666	0.0890	0.0885	0.0723	0.1083	0.0524	0.0976	0.0871	0.1143	0.0743
0.8251	0.1122	0.0726	0.1072	0.0576	0.1172	0.1082	0.1383	0.0658	0.1000	0.0798	0.0864	0.0908	0.0797	0.0926	0.1033	0.1102	0.0997	0.1482	0.0957
0.8356	0.8483	0.0796	0.1161	0.0710	0.1315	0.1426	0.1323	0.0818	0.0707	0.0670	0.0595	0.1048	0.0663	0.0937	0.0767	0.1131	0.1394	0.1351	0.0875
0.8181	0.8336	0.8087	0.1504	0.0995	0.1525	0.1777	0.0975	0.1435	0.0585	0.0599	0.0563	0.0873	0.0830	0.0795	0.1108	0.1472	0.1068	0.1286	0.0743
0.8203	0.8404	0.8436	0.8364	0.0700	0.1068	0.1246	0.0773	0.1236	0.0375	0.0659	0.0521	0.0849	0.0760	0.0930	0.0770	0.1111	0.0875	0.1255	0.1012
0.8278	0.8113	0.8264	0.8375	0.8216	0.1582	0.0980	0.0689	0.1044	0.0559	0.0716	0.0700	0.1016	0.0614	0.0902	0.0770	0.1049	0.0923	0.0837	0.1024
0.8254	0.8446	0.8402	0.8460	0.8509	0.8316	0.0935	0.0805	0.0795	0.0575	0.0499	0.0711	0.0610	0.0829	0.0978	0.0792	0.1077	0.1061	0.1406	0.0625
0.7973	0.8050	0.8291	0.8315	0.8201	0.8492	0.8164	0.0966	0.1139	0.0808	0.0499	0.0871	0.0620	0.0720	0.1085	0.0653	0.1221	0.1116	0.1350	0.1018
0.8057	0.8302	0.8360	0.8314	0.8356	0.8374	0.8532	0.7855	0.0950	0.0643	0.0603	0.0784	0.0669	0.0836	0.0923	0.0666	0.0952	0.1046	0.1100	0.0785
0.7867	0.8097	0.8336	0.8128	0.7878	0.8208	0.8319	0.8142	0.8092	0.0816	0.0715	0.0745	0.0670	0.0897	0.1213	0.0690	0.1025	0.1148	0.1243	0.0844
0.8064	0.8273	0.8331	0.8298	0.8257	0.8458	0.8306	0.8205	0.8637	0.7969	0.0613	0.0747	0.0717	0.0850	0.0951	0.0677	0.1206	0.1118	0.1233	0.1017
0.7963	0.8131	0.8407	0.7987	0.8265	0.8322	0.8293	0.8287	0.8489	0.8395	0.8031	0.0869	0.0768	0.0814	0.1066	0.0587	0.1236	0.1216	0.1424	0.0894
0.7986	0.7896	0.8230	0.7921	0.8165	0.8277	0.8312	0.8120	0.8370	0.8387	0.8360	0.7894	0.0702	0.1012	0.1289	0.0556	0.1315	0.1288	0.1151	0.0943
0.7971	0.7997	0.8205	0.8190	0.8278	0.8343	0.8364	0.8182	0.8377	0.8203	0.8404	0.8417	0.7844	0.1085	0.1165	0.0805	0.1237	0.1343	0.1247	0.1053
0.7903	0.8206	0.8194	0.8173	0.8103	0.8307	0.8307	0.8109	0.8434	0.8237	0.8338	0.8474	0.8136	0.8241	0.1098	0.0772	0.1345	0.1340	0.1156	0.0821
0.7819	0.8161	0.8111	0.8090	0.8174	0.8258	0.8378	0.8018	0.8325	0.8276	0.8171	0.8243	0.8312	0.8255	0.8517	0.0658	0.1202	0.1227	0.1291	0.0957
0.7807	0.8159	0.8169	0.8054	0.8139	0.8246	0.8272	0.7853	0.8331	0.8119	0.8111	0.8023	0.8358	0.8275	0.8321	0.7999	0.0992	0.1114	0.1391	0.0802
0.7854	0.7854	0.8076	0.8116	0.8190	0.8177	0.8256	0.7916	0.8211	0.8186	0.8315	0.8180	0.8312	0.8227	0.8290	0.8192	0.7936	0.1204	0.1219	0.0967
0.7716	0.7946	0.7920	0.8053	0.8029	0.8094	0.8246	0.7756	0.8143	0.8028	0.8076	0.8152	0.8224	0.8051	0.8282	0.8133	0.8265	0.8246	0.1260	0.0971
0.7702	0.7946	0.7990	0.7934	0.8039	0.8169	0.8315	0.7842	0.8204	0.8168	0.8190	0.8122	0.8188	0.8172	0.8259	0.8255	0.8395	0.8425	0.8297	0.1018
0.7651	0.7853	0.7984	0.8025	0.8129	0.8081	0.8199	0.7710	0.8215	0.8105	0.8217	0.8109	0.8259	0.8171	0.8270	0.8108	0.8231	0.8445	0.8256	0.8146

```
Final Accuracy: 0.8108
Backward: -0.0039
Forward:  0.0070
```

B.2 MNIST rotations

B.2.1 Model *single*

0.1054	0.0950	0.0775	0.0711	0.0971	0.0872	0.1043	0.1012	0.0765	0.0775	0.0777	0.0777	0.0891	0.0961	0.0904	0.0819	0.0825	0.0759	0.0845	0.1089
0.3016	0.2605	0.2517	0.2407	0.2213	0.1658	0.1573	0.1521	0.0958	0.0860	0.0911	0.0876	0.1135	0.1182	0.1101	0.1107	0.1028	0.1195	0.1479	0.1673
0.4327	0.4485	0.3883	0.3712	0.3235	0.2303	0.2052	0.1903	0.1332	0.1260	0.1294	0.1189	0.1364	0.1365	0.1220	0.1196	0.1097	0.1175	0.1581	0.1759
0.5016	0.5674	0.5277	0.5097	0.4605	0.3262	0.2656	0.2301	0.1639	0.1515	0.1515	0.1369	0.1479	0.1474	0.1323	0.1287	0.1142	0.1220	0.1547	0.1719
0.5051	0.5859	0.5807	0.5705	0.5369	0.3994	0.3316	0.2902	0.1865	0.1660	0.1569	0.1417	0.1432	0.1452	0.1306	0.1243	0.1195	0.1378	0.1651	0.1844
0.5099	0.6172	0.6350	0.6356	0.6185	0.4822	0.3838	0.3328	0.2069	0.1797	0.1676	0.1486	0.1342	0.1368	0.1244	0.1299	0.1301	0.1518	0.1821	0.2071
0.4926	0.6225	0.6658	0.6889	0.6993	0.6129	0.5047	0.4487	0.2784	0.2144	0.2050	0.1663	0.1463	0.1457	0.1369	0.1376	0.1372	0.1493	0.1710	0.1928
0.4474	0.5830	0.6326	0.6644	0.6901	0.6657	0.6109	0.5578	0.3849	0.2929	0.2760	0.2107	0.1651	0.1582	0.1362	0.1421	0.1442	0.1671	0.1871	0.2119
0.4473	0.5827	0.6361	0.6656	0.7091	0.7239	0.7090	0.6798	0.5201	0.4113	0.3770	0.2663	0.1885	0.1778	0.1513	0.1467	0.1377	0.1562	0.1746	0.2031
0.3929	0.5233	0.5923	0.6246	0.6793	0.7091	0.7193	0.7015	0.6373	0.5319	0.4929	0.3709	0.2369	0.2113	0.1687	0.1435	0.1386	0.1527	0.1764	0.2057
0.3651	0.4899	0.5613	0.5961	0.6573	0.6910	0.7098	0.7034	0.6882	0.6271	0.6026	0.4852	0.3259	0.2970	0.2351	0.1711	0.1537	0.1572	0.1765	0.2026
0.3400	0.4652	0.5347	0.5774	0.6487	0.7052	0.7386	0.7402	0.7359	0.6933	0.6721	0.5581	0.3909	0.3648	0.2847	0.1953	0.1647	0.1580	0.1675	0.1928
0.3017	0.4081	0.4748	0.5225	0.5949	0.6729	0.7110	0.7154	0.7395	0.7266	0.7152	0.6482	0.5002	0.4643	0.3809	0.2614	0.2113	0.1882	0.1866	0.2087
0.2956	0.3872	0.4404	0.4817	0.5481	0.6431	0.6852	0.6968	0.7289	0.7413	0.7375	0.7117	0.6267	0.5918	0.5126	0.3757	0.3027	0.2383	0.2092	0.2205
0.2839	0.3559	0.4004	0.4392	0.5083	0.6140	0.6585	0.6708	0.7114	0.7384	0.7412	0.7415	0.7066	0.6920	0.6191	0.4779	0.3772	0.2798	0.2224	0.2332
0.2716	0.3341	0.3630	0.4009	0.4662	0.5708	0.6176	0.6381	0.6944	0.7363	0.7448	0.7686	0.7626	0.7572	0.7148	0.5815	0.4717	0.3385	0.2407	0.2421
0.2643	0.3250	0.3448	0.3814	0.4381	0.5356	0.5834	0.5995	0.6529	0.7047	0.7073	0.7506	0.7769	0.7777	0.7546	0.6862	0.5957	0.4513	0.3045	0.2969
0.2616	0.3216	0.3296	0.3593	0.4088	0.4953	0.5299	0.5527	0.6058	0.6575	0.6654	0.7104	0.7520	0.7517	0.7531	0.7313	0.6920	0.5751	0.4154	0.4008
0.2398	0.2977	0.3064	0.3312	0.3737	0.4450	0.4804	0.5012	0.5689	0.6212	0.6293	0.6881	0.7452	0.7540	0.7667	0.7669	0.7491	0.6716	0.5180	0.4960
0.2292	0.2815	0.2844	0.3063	0.3472	0.4059	0.4377	0.4609	0.5282	0.5842	0.5939	0.6574	0.7143	0.7251	0.7390	0.7595	0.7595	0.7277	0.6419	0.6095
0.2237	0.2817	0.2925	0.3227	0.3532	0.3987	0.4260	0.4504	0.5051	0.5444	0.5513	0.6090	0.6741	0.6862	0.7052	0.7426	0.7531	0.7526	0.7065	0.6839

```
Final Accuracy: 0.5331
Backward: -0.0851
Forward:  0.4196
```

B.2.2 Model *independent*

0.1024	0.0928	0.0982	0.0772	0.0963	0.1077	0.1116	0.0912	0.0807	0.1142	0.0890	0.1225	0.1193	0.0874	0.0927	0.0961	0.1101	0.1042	0.0880	0.1844
0.3296	0.0928	0.0982	0.0772	0.0963	0.1077	0.1116	0.0912	0.0807	0.1142	0.0890	0.1225	0.1193	0.0874	0.0927	0.0961	0.1101	0.1042	0.0880	0.1844
0.3296	0.4522	0.0982	0.0772	0.0963	0.1077	0.1116	0.0912	0.0807	0.1142	0.0890	0.1225	0.1193	0.0874	0.0927	0.0961	0.1101	0.1042	0.0880	0.1844
0.3296	0.4522	0.4249	0.0772	0.0963	0.1077	0.1116	0.0912	0.0807	0.1142	0.0890	0.1225	0.1193	0.0874	0.0927	0.0961	0.1101	0.1042	0.0880	0.1844
0.3296	0.4522	0.4249	0.5594	0.0963	0.1077	0.1116	0.0912	0.0807	0.1142	0.0890	0.1225	0.1193	0.0874	0.0927	0.0961	0.1101	0.1042	0.0880	0.1844
0.3296	0.4522	0.4249	0.5594	0.5011	0.1077	0.1116	0.0912	0.0807	0.1142	0.0890	0.1225	0.1193	0.0874	0.0927	0.0961	0.1101	0.1042	0.0880	0.1844
0.3296	0.4522	0.4249	0.5594	0.5011	0.5186	0.1116	0.0912	0.0807	0.1142	0.0890	0.1225	0.1193	0.0874	0.0927	0.0961	0.1101	0.1042	0.0880	0.1844
0.3296	0.4522	0.4249	0.5594	0.5011	0.5186	0.5530	0.0912	0.0807	0.1142	0.0890	0.1225	0.1193	0.0874	0.0927	0.0961	0.1101	0.1042	0.0880	0.1844
0.3296	0.4522	0.4249	0.5594	0.5011	0.5186	0.5530	0.5384	0.0807	0.1142	0.0890	0.1225	0.1193	0.0874	0.0927	0.0961	0.1101	0.1042	0.0880	0.1844
0.3296	0.4522	0.4249	0.5594	0.5011	0.5186	0.5530	0.5384	0.6028	0.1142	0.0890	0.1225	0.1193	0.0874	0.0927	0.0961	0.1101	0.1042	0.0880	0.1844
0.3296	0.4522	0.4249	0.5594	0.5011	0.5186	0.5530	0.5384	0.6028	0.6337	0.0890	0.1225	0.1193	0.0874	0.0927	0.0961	0.1101	0.1042	0.0880	0.1844
0.3296	0.4522	0.4249	0.5594	0.5011	0.5186	0.5530	0.5384	0.6028	0.6337	0.6506	0.1225	0.1193	0.0874	0.0927	0.0961	0.1101	0.1042	0.0880	0.1844
0.3296	0.4522	0.4249	0.5594	0.5011	0.5186	0.5530	0.5384	0.6028	0.6337	0.6506	0.5542	0.1193	0.0874	0.0927	0.0961	0.1101	0.1042	0.0880	0.1844
0.3296	0.4522	0.4249	0.5594	0.5011	0.5186	0.5530	0.5384	0.6028	0.6337	0.6506	0.5542	0.6052	0.0874	0.0927	0.0961	0.1101	0.1042	0.0880	0.1844
0.3296	0.4522	0.4249	0.5594	0.5011	0.5186	0.5530	0.5384	0.6028	0.6337	0.6506	0.5542	0.6052	0.6204	0.0927	0.0961	0.1101	0.1042	0.0880	0.1844
0.3296	0.4522	0.4249	0.5594	0.5011	0.5186	0.5530	0.5384	0.6028	0.6337	0.6506	0.5542	0.6052	0.6204	0.5810	0.0961	0.1101	0.1042	0.0880	0.1844
0.3296	0.4522	0.4249	0.5594	0.5011	0.5186	0.5530	0.5384	0.6028	0.6337	0.6506	0.5542	0.6052	0.6204	0.5810	0.6296	0.1101	0.1042	0.0880	0.1844
0.3296	0.4522	0.4249	0.5594	0.5011	0.5186	0.5530	0.5384	0.6028	0.6337	0.6506	0.5542	0.6052	0.6204	0.5810	0.6296	0.6437	0.1042	0.0880	0.1844
0.3296	0.4522	0.4249	0.5594	0.5011	0.5186	0.5530	0.5384	0.6028	0.6337	0.6506	0.5542	0.6052	0.6204	0.5810	0.6296	0.6437	0.6333	0.0880	0.1844
0.3296	0.4522	0.4249	0.5594	0.5011	0.5186	0.5530	0.5384	0.6028	0.6337	0.6506	0.5542	0.6052	0.6204	0.5810	0.6296	0.6437	0.6333	0.5625	0.1844
0.3296	0.4522	0.4249	0.5594	0.5011	0.5186	0.5530	0.5384	0.6028	0.6337	0.6506	0.5542	0.6052	0.6204	0.5810	0.6296	0.6437	0.6333	0.5625	0.5897

```
Final Accuracy: 0.5592
Backward: 0.0000
Forward:  0.0000
```

B.2.3 Model *multitask*

0.0994	0.0860	0.0955	0.0861	0.1097	0.0943	0.0837	0.0631	0.1423	0.0886	0.0944	0.0902	0.0945	0.0460	0.0807	0.0842	0.1008	0.0832	0.0670	0.1206
0.7037	0.1454	0.1322	0.1161	0.1673	0.1322	0.1188	0.1351	0.1250	0.0730	0.1289	0.1746	0.1010	0.1142	0.1324	0.0980	0.1281	0.0568	0.1018	0.1393
0.7529	0.8124	0.1234	0.1033	0.1248	0.1354	0.1227	0.1096	0.1242	0.0638	0.1293	0.1142	0.0578	0.1447	0.0690	0.0720	0.1332	0.0708	0.0851	0.1611
0.7238	0.8241	0.8116	0.1009	0.0904	0.1221	0.0900	0.1048	0.1274	0.0893	0.1029	0.1016	0.0965	0.1665	0.0819	0.0949	0.1232	0.0779	0.1170	0.1383
0.7157	0.8100	0.8111	0.8185	0.1169	0.1668	0.1123	0.1079	0.1197	0.0769	0.1252	0.1191	0.0592	0.1637	0.0919	0.0758	0.1014	0.0719	0.0896	0.1583
0.7297	0.7870	0.7969	0.7973	0.7930	0.1581	0.1131	0.0991	0.1025	0.0783	0.1275	0.1033	0.0519	0.1611	0.0477	0.0773	0.0922	0.0725	0.0850	0.1748
0.7572	0.8018	0.8101	0.8352	0.8184	0.8089	0.1281	0.1137	0.0977	0.0633	0.1094	0.1182	0.0637	0.1636	0.0664	0.0690	0.0824	0.0640	0.0722	0.1727
0.7115	0.8108	0.7983	0.8334	0.8268	0.7688	0.6244	0.1036	0.1297	0.0684	0.1096	0.1274	0.0647	0.1888	0.0812	0.0925	0.0736	0.0764	0.0944	0.1414
0.6394	0.7935	0.8044	0.8352	0.8189	0.7716	0.6025	0.7916	0.1324	0.1015	0.1048	0.1526	0.0472	0.1552	0.1331	0.0841	0.1167	0.0667	0.1108	0.1309
0.6828	0.7672	0.7723	0.7884	0.7703	0.7928	0.6413	0.7675	0.6567	0.1052	0.1405	0.1232	0.0453	0.1721	0.0493	0.0871	0.1113	0.0706	0.0996	0.1433
0.6527	0.7833	0.7534	0.8047	0.8142	0.8100	0.5581	0.8177	0.7280	0.7882	0.1079	0.0965	0.0727	0.1813	0.0437	0.0822	0.1132	0.0804	0.1156	0.1444
0.7025	0.8099	0.7801	0.8184	0.8141	0.8144	0.6464	0.8225	0.7073	0.7994	0.8022	0.1379	0.0581	0.1691	0.0549	0.0695	0.0879	0.0667	0.1082	0.1408
0.6891	0.8047	0.7578	0.8156	0.8284	0.8170	0.6119	0.8113	0.7108	0.7984	0.7862	0.7783	0.0515	0.2050	0.0435	0.0673	0.0862	0.0690	0.0992	0.1421
0.6896	0.8139	0.7808	0.8165	0.8084	0.8124	0.6120	0.7977	0.6845	0.8064	0.8031	0.7654	0.7931	0.1906	0.0305	0.0868	0.0760	0.0843	0.1062	0.1415
0.6968	0.8260	0.8018	0.8330	0.8217	0.8160	0.5947	0.8019	0.6695	0.8163	0.8245	0.7641	0.8115	0.8400	0.0470	0.0787	0.0846	0.0812	0.1141	0.1408
0.6929	0.8089	0.7928	0.8336	0.8350	0.8190	0.5982	0.8254	0.6607	0.7922	0.8089	0.7676	0.8085	0.8408	0.8345	0.0685	0.0688	0.0749	0.1057	0.1445
0.6865	0.8065	0.8060	0.8444	0.8238	0.7638	0.6196	0.7680	0.6764	0.7842	0.8159	0.7552	0.8246	0.8225	0.7914	0.7979	0.0649	0.0811	0.1017	0.1306
0.6421	0.7976	0.7901	0.8429	0.8324	0.7848	0.5884	0.7797	0.6812	0.7569	0.8153	0.7502	0.8291	0.8280	0.7991	0.8103	0.7662	0.0659	0.0998	0.1278
0.6525	0.8016	0.8099	0.8342	0.8240	0.7824	0.5708	0.7608	0.6382	0.7662	0.8041	0.7515	0.8086	0.8379	0.8003	0.8007	0.7882	0.8158	0.1007	0.1199
0.7009	0.8040	0.8061	0.8284	0.8254	0.8017	0.6039	0.7591	0.6091	0.7735	0.7913	0.7457	0.7739	0.8221	0.8286	0.8243	0.7836	0.8303	0.8355	0.1223
0.6828	0.7913	0.8114	0.8280	0.8169	0.7699	0.5627	0.7438	0.6163	0.7257	0.7650	0.7196	0.7736	0.8099	0.7932	0.7948	0.7864	0.8141	0.8368	0.8260

```
Final Accuracy: 0.7634
Backward: -0.0215
Forward:  0.0180
```

B.2.4 Model *EWC*

0.1054	0.0950	0.0775	0.0711	0.0971	0.0872	0.1043	0.1012	0.0765	0.0775	0.0777	0.0777	0.0891	0.0961	0.0904	0.0819	0.0825	0.0759	0.0845	0.1089
0.5578	0.5141	0.4132	0.3846	0.3325	0.2314	0.2002	0.1866	0.1475	0.1442	0.1414	0.1443	0.1620	0.1645	0.1487	0.1508	0.1366	0.1627	0.1800	0.1997
0.6745	0.7020	0.6272	0.5959	0.5162	0.3555	0.2670	0.2382	0.1815	0.1632	0.1579	0.1474	0.1369	0.1354	0.1243	0.1191	0.1190	0.1395	0.1734	0.1873
0.6725	0.7637	0.7499	0.7273	0.6515	0.4768	0.3597	0.3087	0.2157	0.1818	0.1765	0.1600	0.1379	0.1430	0.1315	0.1328	0.1318	0.1521	0.1701	0.1888
0.6603	0.7738	0.7946	0.7882	0.7459	0.5655	0.4267	0.3644	0.2182	0.1716	0.1690	0.1397	0.1244	0.1261	0.1247	0.1329	0.1259	0.1460	0.1577	0.1805
0.5876	0.7243	0.7763	0.7856	0.7792	0.6415	0.5108	0.4329	0.2569	0.1963	0.1901	0.1522	0.1341	0.1294	0.1228	0.1339	0.1323	0.1645	0.1767	0.1957
0.5196	0.6928	0.7759	0.8058	0.8286	0.7815	0.6783	0.6125	0.3956	0.2797	0.2675	0.2043	0.1685	0.1614	0.1423	0.1461	0.1343	0.1500	0.1589	0.1752
0.4829	0.6455	0.7389	0.7759	0.8132	0.8279	0.7894	0.7402	0.5513	0.4112	0.3771	0.2597	0.1898	0.1810	0.1612	0.1524	0.1366	0.1447	0.1508	0.1693
0.4420	0.6062	0.6950	0.7342	0.7862	0.8245	0.8114	0.7929	0.6471	0.5173	0.4752	0.3229	0.2208	0.2078	0.1732	0.1541	0.1362	0.1479	0.1611	0.1799
0.3811	0.5290	0.6056	0.6531	0.7268	0.7934	0.8233	0.8137	0.7729	0.6687	0.6370	0.4764	0.2999	0.2730	0.2166	0.1691	0.1438	0.1420	0.1401	0.1597
0.3320	0.4673	0.5428	0.5896	0.6660	0.7467	0.7842	0.7838	0.7911	0.7469	0.7237	0.6009	0.4185	0.3873	0.3026	0.2117	0.1687	0.1573	0.1464	0.1613
0.3180	0.4420	0.5151	0.5612	0.6402	0.7358	0.7806	0.7893	0.8126	0.7884	0.7729	0.6612	0.4702	0.4314	0.3348	0.2221	0.1707	0.1463	0.1286	0.1431
0.3040	0.3876	0.4448	0.4813	0.5570	0.6750	0.7250	0.7467	0.7985	0.8135	0.8154	0.7615	0.6036	0.5703	0.4731	0.3191	0.2527	0.2040	0.1698	0.1738
0.2536	0.3154	0.3543	0.3822	0.4564	0.5513	0.6163	0.6381	0.7178	0.7660	0.7833	0.7911	0.7252	0.6898	0.6065	0.4449	0.3478	0.2501	0.1896	0.1873
0.2701	0.3277	0.3542	0.3768	0.4368	0.5059	0.5660	0.5879	0.6610	0.7273	0.7457	0.7882	0.7805	0.7738	0.7015	0.5461	0.4262	0.2980	0.2053	0.2032
0.2577	0.3210	0.3496	0.3734	0.4318	0.4901	0.5380	0.5599	0.6570	0.7367	0.7538	0.8056	0.8237	0.8145	0.7816	0.6384	0.5036	0.3424	0.2178	0.2060
0.2814	0.3644	0.3876	0.4145	0.4621	0.5001	0.5232	0.5358	0.5967	0.6521	0.6710	0.7216	0.7698	0.7700	0.7585	0.7122	0.6292	0.4626	0.2977	0.2793
0.3102	0.3832	0.3906	0.4065	0.4439	0.4567	0.4676	0.4726	0.5307	0.5990	0.6125	0.6780	0.7289	0.7426	0.7493	0.7551	0.7222	0.6061	0.4348	0.3983
0.2952	0.3744	0.3924	0.4076	0.4413	0.4576	0.4560	0.4646	0.5267	0.5862	0.6000	0.6526	0.7121	0.7296	0.7405	0.7701	0.7707	0.7134	0.5630	0.5220
0.2890	0.3731	0.4052	0.4254	0.4634	0.4637	0.4567	0.4622	0.4926	0.5302	0.5487	0.5895	0.6543	0.6715	0.6772	0.7125	0.7429	0.7495	0.7214	0.6880
0.2948	0.3888	0.4307	0.4594	0.4968	0.5012	0.4924	0.4794	0.4967	0.5308	0.5374	0.5720	0.6090	0.6258	0.6281	0.6424	0.6843	0.7124	0.7270	0.7098

```
Final Accuracy: 0.5510
Backward: -0.1918
Forward:  0.5421
```

B.2.5 Model *GEM*

0.1054	0.0950	0.0775	0.0711	0.0971	0.0872	0.1043	0.1012	0.0765	0.0775	0.0777	0.0777	0.0891	0.0961	0.0904	0.0819	0.0825	0.0759	0.0845	0.1089
0.8026	0.7272	0.6119	0.5476	0.4498	0.2781	0.1992	0.1735	0.1242	0.1076	0.1093	0.0922	0.0951	0.0926	0.0824	0.1141	0.1223	0.1401	0.1533	0.1628
0.8343	0.8544	0.7707	0.7146	0.5837	0.3625	0.2553	0.2208	0.1559	0.1290	0.1254	0.1032	0.0967	0.0964	0.1013	0.1285	0.1348	0.1515	0.1662	0.1790
0.8194	0.8642	0.8435	0.8032	0.7200	0.5077	0.3776	0.3291	0.2197	0.1777	0.1613	0.1287	0.1133	0.1137	0.1164	0.1507	0.1558	0.1801	0.1890	0.1959
0.8158	0.8749	0.8801	0.8675	0.8132	0.5877	0.4333	0.3636	0.2202	0.1747	0.1612	0.1297	0.1038	0.1010	0.1035	0.1328	0.1498	0.1736	0.1863	0.1890
0.7771	0.8578	0.8834	0.8802	0.8620	0.7164	0.5555	0.4643	0.2952	0.2170	0.2071	0.1621	0.1276	0.1247	0.1157	0.1386	0.1477	0.1833	0.1872	0.1950
0.7489	0.8461	0.8787	0.8831	0.8901	0.8413	0.7317	0.6621	0.4403	0.2976	0.2680	0.1751	0.1262	0.1178	0.0999	0.1160	0.1311	0.1564	0.1877	0.1937
0.7361	0.8290	0.8652	0.8765	0.8902	0.8795	0.8473	0.8069	0.6278	0.4656	0.4248	0.2618	0.1809	0.1551	0.1232	0.1236	0.1368	0.1634	0.1868	0.1895
0.7434	0.8262	0.8606	0.8688	0.8860	0.8891	0.8736	0.8569	0.7232	0.5661	0.5149	0.3414	0.2185	0.1881	0.1499	0.1382	0.1469	0.1747	0.1923	0.1893
0.7220	0.8078	0.8412	0.8516	0.8696	0.8748	0.8783	0.8711	0.8395	0.7461	0.6960	0.5176	0.3319	0.2970	0.2266	0.1699	0.1605	0.1781	0.1916	0.1881
0.7261	0.8050	0.8372	0.8481	0.8572	0.8619	0.8642	0.8621	0.8705	0.8333	0.8024	0.6899	0.4903	0.4386	0.3590	0.2584	0.2194	0.2209	0.2300	0.2296
0.7219	0.8007	0.8259	0.8367	0.8453	0.8530	0.8598	0.8631	0.8726	0.8559	0.8449	0.7515	0.5650	0.5142	0.4191	0.2984	0.2419	0.2181	0.2207	0.2203
0.7122	0.7898	0.8172	0.8245	0.8270	0.8296	0.8360	0.8334	0.8592	0.8727	0.8710	0.8305	0.6969	0.6567	0.5505	0.3949	0.3139	0.2568	0.2383	0.2447
0.6933	0.7775	0.8141	0.8195	0.8279	0.8210	0.8219	0.8184	0.8388	0.8631	0.8692	0.8661	0.8220	0.8042	0.7324	0.5694	0.4590	0.3236	0.2713	0.2647
0.6929	0.7726	0.8128	0.8175	0.8249	0.8209	0.8197	0.8173	0.8319	0.8584	0.8658	0.8741	0.8604	0.8522	0.8037	0.6557	0.5366	0.3792	0.3001	0.2880
0.6934	0.7710	0.8146	0.8242	0.8324	0.8260	0.8221	0.8167	0.8248	0.8476	0.8553	0.8794	0.8829	0.8834	0.8658	0.7739	0.6539	0.4869	0.3293	0.2920
0.6821	0.7620	0.8080	0.8185	0.8266	0.8167	0.8133	0.8114	0.8194	0.8374	0.8441	0.8618	0.8754	0.8766	0.8780	0.8498	0.7666	0.6261	0.4324	0.3821
0.6662	0.7532	0.8038	0.8146	0.8255	0.8202	0.8122	0.8080	0.8119	0.8286	0.8296	0.8459	0.8627	0.8701	0.8764	0.8763	0.8432	0.7451	0.5474	0.4764
0.6378	0.7446	0.7952	0.8095	0.8214	0.8206	0.8162	0.8092	0.8089	0.8190	0.8205	0.8323	0.8426	0.8504	0.8627	0.8795	0.8690	0.8354	0.6909	0.6328
0.5755	0.7077	0.7788	0.8010	0.8121	0.8213	0.8188	0.8101	0.8086	0.8211	0.8191	0.8235	0.8251	0.8324	0.8376	0.8544	0.8612	0.8730	0.8401	0.8021
0.5671	0.6978	0.7708	0.7949	0.8101	0.8163	0.8111	0.8003	0.8064	0.8201	0.8205	0.8241	0.8244	0.8249	0.8242	0.8432	0.8586	0.8786	0.8647	0.8440

```
Final Accuracy: 0.8051
Backward: -0.0387
Forward:  0.6412
```

B.3 CIFAR-100 class-incremental

B.3.1 Model *independent*

0.2000	0.2000	0.2000	0.1980	0.2000	0.1980	0.1980	0.2000	0.1980	0.2000	0.2000	0.2000	0.2000	0.2000	0.1980	0.2000	0.1980	0.2000	0.2000	0.2000
0.3580	0.2000	0.2000	0.1980	0.2000	0.1980	0.1980	0.2000	0.1980	0.2000	0.2000	0.2000	0.2000	0.2000	0.1980	0.2000	0.1980	0.2000	0.2000	0.2000
0.3580	0.3000	0.2000	0.1980	0.2000	0.1980	0.1980	0.2000	0.1980	0.2000	0.2000	0.2000	0.2000	0.2000	0.1980	0.2000	0.1980	0.2000	0.2000	0.2000
0.3580	0.3000	0.2640	0.1980	0.2000	0.1980	0.1980	0.2000	0.1980	0.2000	0.2000	0.2000	0.2000	0.2000	0.1980	0.2000	0.1980	0.2000	0.2000	0.2000
0.3580	0.3000	0.2640	0.3500	0.2000	0.1980	0.1980	0.2000	0.1980	0.2000	0.2000	0.2000	0.2000	0.2000	0.1980	0.2000	0.1980	0.2000	0.2000	0.2000
0.3580	0.3000	0.2640	0.3500	0.5100	0.1980	0.1980	0.2000	0.1980	0.2000	0.2000	0.2000	0.2000	0.2000	0.1980	0.2000	0.1980	0.2000	0.2000	0.2000
0.3580	0.3000	0.2640	0.3500	0.5100	0.2260	0.1980	0.2000	0.1980	0.2000	0.2000	0.2000	0.2000	0.2000	0.1980	0.2000	0.1980	0.2000	0.2000	0.2000
0.3580	0.3000	0.2640	0.3500	0.5100	0.2260	0.4180	0.2000	0.1980	0.2000	0.2000	0.2000	0.2000	0.2000	0.1980	0.2000	0.1980	0.2000	0.2000	0.2000
0.3580	0.3000	0.2640	0.3500	0.5100	0.2260	0.4180	0.3720	0.1980	0.2000	0.2000	0.2000	0.2000	0.2000	0.1980	0.2000	0.1980	0.2000	0.2000	0.2000
0.3580	0.3000	0.2640	0.3500	0.5100	0.2260	0.4180	0.3720	0.2880	0.2000	0.2000	0.2000	0.2000	0.2000	0.1980	0.2000	0.1980	0.2000	0.2000	0.2000
0.3580	0.3000	0.2640	0.3500	0.5100	0.2260	0.4180	0.3720	0.2880	0.3640	0.2000	0.2000	0.2000	0.2000	0.1980	0.2000	0.1980	0.2000	0.2000	0.2000
0.3580	0.3000	0.2640	0.3500	0.5100	0.2260	0.4180	0.3720	0.2880	0.3640	0.4820	0.2000	0.2000	0.2000	0.1980	0.2000	0.1980	0.2000	0.2000	0.2000
0.3580	0.3000	0.2640	0.3500	0.5100	0.2260	0.4180	0.3720	0.2880	0.3640	0.4820	0.3740	0.2000	0.2000	0.1980	0.2000	0.1980	0.2000	0.2000	0.2000
0.3580	0.3000	0.2640	0.3500	0.5100	0.2260	0.4180	0.3720	0.2880	0.3640	0.4820	0.3740	0.4780	0.2000	0.1980	0.2000	0.1980	0.2000	0.2000	0.2000
0.3580	0.3000	0.2640	0.3500	0.5100	0.2260	0.4180	0.3720	0.2880	0.3640	0.4820	0.3740	0.4780	0.4240	0.1980	0.2000	0.1980	0.2000	0.2000	0.2000
0.3580	0.3000	0.2640	0.3500	0.5100	0.2260	0.4180	0.3720	0.2880	0.3640	0.4820	0.3740	0.4780	0.4240	0.3880	0.2000	0.1980	0.2000	0.2000	0.2000
0.3580	0.3000	0.2640	0.3500	0.5100	0.2260	0.4180	0.3720	0.2880	0.3640	0.4820	0.3740	0.4780	0.4240	0.3880	0.3800	0.1980	0.2000	0.2000	0.2000
0.3580	0.3000	0.2640	0.3500	0.5100	0.2260	0.4180	0.3720	0.2880	0.3640	0.4820	0.3740	0.4780	0.4240	0.3880	0.3800	0.3580	0.2000	0.2000	0.2000
0.3580	0.3000	0.2640	0.3500	0.5100	0.2260	0.4180	0.3720	0.2880	0.3640	0.4820	0.3740	0.4780	0.4240	0.3880	0.3800	0.3580	0.3300	0.2000	0.2000
0.3580	0.3000	0.2640	0.3500	0.5100	0.2260	0.4180	0.3720	0.2880	0.3640	0.4820	0.3740	0.4780	0.4240	0.3880	0.3800	0.3580	0.3300	0.3480	0.2000
0.3580	0.3000	0.2640	0.3500	0.5100	0.2260	0.4180	0.3720	0.2880	0.3640	0.4820	0.3740	0.4780	0.4240	0.3880	0.3800	0.3580	0.3300	0.3480	0.3900

```
Final Accuracy: 0.3701
Backward: 0.0000
Forward:  0.0000
```

B.3.2 Model *single*

0.2000	0.2000	0.2000	0.2000	0.1980	0.2000	0.1980	0.1980	0.2000	0.2000	0.2000	0.2000	0.2000	0.2000	0.2000	0.2000	0.2000	0.1980	0.2000	0.2000
0.4240	0.1960	0.2140	0.1920	0.2020	0.2000	0.2560	0.2220	0.2000	0.2220	0.1980	0.1740	0.2240	0.2780	0.2040	0.2320	0.3060	0.1840	0.1460	0.1880
0.3520	0.3260	0.1980	0.1860	0.2040	0.2020	0.1840	0.2640	0.2000	0.1980	0.2120	0.1120	0.2580	0.1940	0.1420	0.2300	0.2400	0.1860	0.1700	0.1380
0.3160	0.2960	0.3600	0.1160	0.1700	0.2000	0.2560	0.2340	0.2000	0.1760	0.1880	0.2080	0.3380	0.2260	0.1360	0.2400	0.2660	0.2220	0.2080	0.1620
0.3120	0.2520	0.3740	0.3880	0.2260	0.2020	0.3300	0.1560	0.2080	0.1980	0.1900	0.2080	0.3620	0.1900	0.1620	0.2380	0.1280	0.1600	0.2460	0.2140
0.2580	0.2720	0.3600	0.3780	0.6580	0.2000	0.2580	0.1280	0.2760	0.2320	0.2300	0.1960	0.2720	0.1800	0.1920	0.2240	0.1640	0.2220	0.2620	0.1600
0.2720	0.2900	0.2840	0.2780	0.5000	0.3060	0.2060	0.2360	0.2880	0.2000	0.1940	0.2080	0.2160	0.1540	0.2040	0.2440	0.3160	0.2400	0.1680	0.2180
0.3020	0.2800	0.2940	0.3280	0.4460	0.2380	0.5120	0.2460	0.2900	0.1960	0.2040	0.2120	0.2420	0.1260	0.2200	0.2440	0.2060	0.2240	0.1140	0.2220
0.2900	0.2860	0.3500	0.3060	0.5460	0.2860	0.4840	0.4620	0.3040	0.2260	0.2120	0.2040	0.2040	0.1220	0.2260	0.1860	0.1980	0.2020	0.1060	0.1820
0.2480	0.2320	0.2780	0.2400	0.5400	0.2660	0.4600	0.3540	0.4600	0.2020	0.2180	0.2220	0.2980	0.1040	0.1880	0.1780	0.2320	0.1700	0.1760	0.2260
0.3180	0.2420	0.3220	0.2660	0.5060	0.2640	0.4960	0.4060	0.4380	0.5720	0.1980	0.1360	0.3380	0.1920	0.1440	0.2080	0.2260	0.1820	0.1640	0.2500
0.3540	0.2740	0.2840	0.2320	0.3980	0.2460	0.4380	0.3160	0.3400	0.3860	0.6980	0.1120	0.4040	0.1480	0.2020	0.1720	0.2180	0.1760	0.1500	0.2320
0.3260	0.3280	0.2780	0.2600	0.4860	0.3000	0.5040	0.3640	0.4160	0.5900	0.6700	0.5320	0.2720	0.2300	0.1840	0.1900	0.2080	0.1600	0.1880	0.2400
0.4060	0.2700	0.3360	0.2680	0.5860	0.2800	0.4880	0.3740	0.3860	0.5460	0.6480	0.3600	0.6980	0.1600	0.1120	0.2500	0.1780	0.2160	0.1580	0.1840
0.3920	0.2620	0.3280	0.2880	0.4820	0.2560	0.5000	0.4440	0.4220	0.5680	0.6120	0.4180	0.6600	0.5320	0.1200	0.1860	0.2000	0.1720	0.1380	0.1740
0.4160	0.2680	0.3500	0.2360	0.5560	0.2360	0.4760	0.3820	0.2400	0.5380	0.6460	0.3500	0.6200	0.5360	0.7040	0.2600	0.1820	0.1860	0.1340	0.2580
0.3160	0.2740	0.2940	0.2540	0.4740	0.2700	0.5000	0.3360	0.3860	0.4600	0.4720	0.4080	0.6280	0.4660	0.5360	0.5120	0.2180	0.2020	0.1220	0.2380
0.2960	0.2680	0.2860	0.2420	0.5400	0.2820	0.4080	0.3960	0.3120	0.5160	0.5480	0.4280	0.6220	0.5000	0.4940	0.4900	0.5420	0.1680	0.2220	0.2760
0.2740	0.3060	0.2820	0.2500	0.4820	0.2080	0.5160	0.3860	0.4280	0.5000	0.5320	0.3580	0.5600	0.4540	0.4920	0.3800	0.4320	0.6000	0.1240	0.2440
0.3780	0.3140	0.3160	0.2700	0.5480	0.2420	0.4480	0.4260	0.3480	0.5060	0.6440	0.3860	0.6160	0.4620	0.5680	0.4740	0.5220	0.4400	0.6340	0.2560
0.3660	0.2920	0.2980	0.2580	0.5380	0.2360	0.5240	0.4920	0.3800	0.5360	0.6400	0.3860	0.6040	0.4840	0.5620	0.4740	0.5000	0.5020	0.5440	0.7160

```
Final Accuracy: 0.4666
Backward: -0.0652
Forward:  -0.0005
```

B.3.3 Model *EWC*

0.2000	0.2000	0.2000	0.2000	0.1980	0.2000	0.1980	0.1980	0.2000	0.2000	0.2000	0.2000	0.2000	0.2000	0.2000	0.2000	0.2000	0.1980	0.2000	0.2000
0.4240	0.1960	0.2140	0.1920	0.2020	0.2000	0.2560	0.2220	0.2000	0.2220	0.1980	0.1740	0.2240	0.2780	0.2040	0.2320	0.3060	0.1840	0.1460	0.1880
0.3500	0.3340	0.1980	0.1960	0.2200	0.2000	0.2200	0.2580	0.2000	0.2060	0.1800	0.1340	0.2400	0.1880	0.1380	0.2160	0.2500	0.1940	0.1860	0.1500
0.3100	0.2960	0.3680	0.1560	0.1880	0.2060	0.3080	0.2400	0.2020	0.2060	0.1420	0.1860	0.2980	0.1700	0.1700	0.1440	0.1880	0.2100	0.1640	0.1520
0.3260	0.2520	0.3320	0.4700	0.1720	0.2100	0.3960	0.2900	0.2100	0.2100	0.1820	0.1720	0.3000	0.2300	0.1620	0.2160	0.2120	0.1960	0.2240	0.1560
0.3260	0.2940	0.3300	0.3440	0.6460	0.2080	0.4140	0.1980	0.2000	0.2120	0.0900	0.2580	0.3580	0.2760	0.1240	0.1040	0.2540	0.2460	0.1660	0.1020
0.3080	0.3060	0.2940	0.2980	0.4720	0.3120	0.2860	0.2840	0.2980	0.2040	0.1260	0.1200	0.1680	0.2120	0.1840	0.2020	0.2580	0.1840	0.2040	0.1580
0.3000	0.2460	0.2740	0.3180	0.5240	0.2280	0.5660	0.2480	0.2420	0.2140	0.2040	0.1660	0.1800	0.2240	0.2040	0.2300	0.2440	0.1960	0.1740	0.2240
0.2960	0.2820	0.2840	0.4060	0.5420	0.2600	0.5180	0.4860	0.2140	0.2240	0.1380	0.2160	0.2420	0.1800	0.1740	0.2440	0.1820	0.1900	0.1520	0.1480
0.2340	0.2820	0.2820	0.3480	0.5620	0.2480	0.4600	0.4080	0.4280	0.2020	0.1620	0.2240	0.2400	0.1960	0.1460	0.2960	0.2140	0.2000	0.2420	0.1200
0.2860	0.3320	0.3080	0.3400	0.4960	0.3080	0.5200	0.3340	0.3740	0.5940	0.1800	0.1980	0.3060	0.2500	0.1000	0.1920	0.1600	0.1900	0.2260	0.1700
0.2760	0.3200	0.2320	0.2520	0.3980	0.2820	0.4720	0.2720	0.2440	0.4380	0.6180	0.2000	0.2160	0.1800	0.1660	0.2980	0.1860	0.2380	0.1180	0.2820
0.2660	0.3340	0.2900	0.3380	0.3620	0.2360	0.4420	0.4220	0.4080	0.4660	0.6200	0.5080	0.2220	0.1720	0.1400	0.1060	0.1500	0.1860	0.1700	0.2200
0.3720	0.3540	0.3100	0.3060	0.5160	0.2900	0.5260	0.3460	0.3520	0.5580	0.6660	0.4380	0.7180	0.2340	0.1120	0.2060	0.1460	0.1820	0.1540	0.2460
0.3780	0.3240	0.3040	0.2860	0.4960	0.2760	0.4660	0.4200	0.3760	0.5420	0.6200	0.3920	0.6620	0.5040	0.2220	0.2620	0.1640	0.1880	0.2080	0.2200
0.3820	0.2860	0.2560	0.3520	0.4380	0.2460	0.4760	0.3720	0.2340	0.4960	0.5860	0.3900	0.6640	0.5200	0.6900	0.1900	0.1720	0.2100	0.2140	0.2340
0.2880	0.3060	0.2680	0.2760	0.4220	0.2760	0.5340	0.3560	0.4100	0.4640	0.5860	0.4160	0.5540	0.4880	0.5820	0.5380	0.2280	0.1900	0.1620	0.1220
0.3000	0.2160	0.2360	0.2740	0.5140	0.2340	0.4840	0.3160	0.3120	0.4820	0.5620	0.3880	0.5680	0.5000	0.5480	0.4620	0.5140	0.1880	0.1700	0.2340
0.2800	0.2040	0.2500	0.2640	0.4340	0.2280	0.4940	0.3080	0.4100	0.4660	0.5120	0.3960	0.4960	0.4500	0.4620	0.4340	0.4220	0.5500	0.1200	0.1500
0.2880	0.3240	0.2720	0.2720	0.3860	0.2740	0.4560	0.3200	0.3860	0.4660	0.5420	0.5220	0.6180	0.4560	0.5540	0.4080	0.4700	0.4300	0.6000	0.1800
0.3080	0.3040	0.3000	0.3940	0.4680	0.2920	0.5260	0.4540	0.4300	0.6020	0.6580	0.4460	0.6480	0.5540	0.5660	0.4600	0.4540	0.4480	0.5940	0.6940

```
Final Accuracy: 0.4800
Backward: -0.0481
Forward:  0.0026
```

B.3.4 Model *iCARL*

0.2000	0.2000	0.2000	0.2000	0.2000	0.2000	0.1980	0.2000	0.2000	0.2000	0.2000	0.2000	0.2000	0.1980	0.1980	0.1980	0.2000	0.2000	0.1980	0.2000
0.4460	0.2000	0.2000	0.2000	0.2000	0.2000	0.1980	0.2000	0.2000	0.2000	0.2000	0.2000	0.2000	0.1980	0.1980	0.1980	0.2000	0.2000	0.1980	0.2000
0.4540	0.3980	0.2000	0.2000	0.2000	0.2000	0.1980	0.2000	0.2000	0.2000	0.2000	0.2000	0.2000	0.1980	0.1980	0.1980	0.2000	0.2000	0.1980	0.2000
0.4260	0.3560	0.3880	0.2000	0.2000	0.2000	0.1980	0.2000	0.2000	0.2000	0.2000	0.2000	0.2000	0.1980	0.1980	0.1980	0.2000	0.2000	0.1980	0.2000
0.4020	0.3400	0.4300	0.4800	0.2000	0.2000	0.1980	0.2000	0.2000	0.2000	0.2000	0.2000	0.2000	0.1980	0.1980	0.1980	0.2000	0.2000	0.1980	0.2000
0.4180	0.3180	0.3840	0.4220	0.5140	0.2000	0.1980	0.2000	0.2000	0.2000	0.2000	0.2000	0.2000	0.1980	0.1980	0.1980	0.2000	0.2000	0.1980	0.2000
0.3960	0.4080	0.3880	0.4480	0.3880	0.3700	0.1980	0.2000	0.2000	0.2000	0.2000	0.2000	0.2000	0.1980	0.1980	0.1980	0.2000	0.2000	0.1980	0.2000
0.4140	0.4300	0.4080	0.4560	0.4180	0.3140	0.4520	0.2000	0.2000	0.2000	0.2000	0.2000	0.2000	0.1980	0.1980	0.1980	0.2000	0.2000	0.1980	0.2000
0.4980	0.4420	0.4400	0.4740	0.4960	0.3860	0.4920	0.4300	0.2000	0.2000	0.2000	0.2000	0.2000	0.1980	0.1980	0.1980	0.2000	0.2000	0.1980	0.2000
0.4380	0.4080	0.4060	0.4460	0.4500	0.3460	0.3920	0.4400	0.4580	0.2000	0.2000	0.2000	0.2000	0.1980	0.1980	0.1980	0.2000	0.2000	0.1980	0.2000
0.4520	0.4300	0.4500	0.4140	0.4540	0.3640	0.3460	0.4500	0.4340	0.6520	0.2000	0.2000	0.2000	0.1980	0.1980	0.1980	0.2000	0.2000	0.1980	0.2000
0.4820	0.4440	0.4020	0.4140	0.4920	0.3220	0.3860	0.4020	0.4000	0.4660	0.7160	0.2000	0.2000	0.1980	0.1980	0.1980	0.2000	0.2000	0.1980	0.2000
0.4580	0.4220	0.4060	0.4360	0.4540	0.3560	0.4380	0.4080	0.4340	0.5300	0.5860	0.4940	0.2000	0.1980	0.1980	0.1980	0.2000	0.2000	0.1980	0.2000
0.5060	0.3780	0.3800	0.4600	0.4240	0.2560	0.4100	0.4740	0.4980	0.4700	0.4900	0.4600	0.7100	0.1980	0.1980	0.1980	0.2000	0.2000	0.1980	0.2000
0.5240	0.3980	0.4200	0.4500	0.5080	0.3260	0.4200	0.4560	0.4260	0.5260	0.5180	0.4280	0.6140	0.5460	0.1980	0.1980	0.2000	0.2000	0.1980	0.2000
0.4880	0.4640	0.4040	0.4000	0.4420	0.3460	0.3720	0.4300	0.4280	0.4580	0.4440	0.4160	0.6420	0.4560	0.6320	0.1980	0.2000	0.2000	0.1980	0.2000
0.4680	0.3880	0.4180	0.4340	0.4820	0.3520	0.4600	0.4000	0.3880	0.4860	0.4360	0.4140	0.5540	0.4320	0.5240	0.5040	0.2000	0.2000	0.1980	0.2000
0.4920	0.4320	0.3980	0.4140	0.4640	0.3340	0.4240	0.3940	0.4620	0.4940	0.6240	0.4400	0.5060	0.4420	0.6140	0.4180	0.5120	0.2000	0.1980	0.2000
0.5180	0.3940	0.4280	0.4900	0.4940	0.3500	0.4360	0.4360	0.4380	0.5380	0.5840	0.3960	0.5880	0.4300	0.5940	0.3560	0.4920	0.5160	0.1980	0.2000
0.4940	0.4440	0.4140	0.4420	0.4460	0.3420	0.4280	0.4120	0.4140	0.4540	0.5160	0.4020	0.5580	0.4360	0.6080	0.4160	0.4840	0.4660	0.7120	0.2000
0.5220	0.4600	0.4620	0.4960	0.4820	0.3840	0.4940	0.4640	0.4060	0.5780	0.6220	0.4460	0.5860	0.4540	0.5620	0.4200	0.5240	0.5280	0.6280	0.6320

```
Final Accuracy: 0.5075
Backward: -0.0206
Forward:  0.0000
```

B.3.5 Model *GEM*

0.2000	0.2000	0.2000	0.2000	0.1980	0.2000	0.1980	0.1980	0.2000	0.2000	0.2000	0.2000	0.2000	0.2000	0.2000	0.2000	0.2000	0.1980	0.2000	0.2000
0.5080	0.1940	0.1720	0.2240	0.2260	0.1980	0.1460	0.2180	0.1960	0.2040	0.2020	0.2440	0.2760	0.1620	0.2480	0.3040	0.2340	0.1980	0.1720	0.2260
0.5040	0.5140	0.1920	0.2040	0.3000	0.1960	0.1260	0.2220	0.1980	0.2000	0.2220	0.2120	0.2920	0.1580	0.2220	0.2440	0.2340	0.1980	0.1840	0.2260
0.5060	0.4760	0.4320	0.1840	0.2340	0.2060	0.1580	0.2260	0.2120	0.1940	0.1960	0.2120	0.3180	0.1760	0.2200	0.1580	0.1880	0.1960	0.2160	0.2680
0.5260	0.4380	0.5220	0.5940	0.2360	0.2060	0.2100	0.1120	0.1640	0.1980	0.1900	0.2180	0.3180	0.1980	0.2020	0.1800	0.1660	0.2000	0.1860	0.2840
0.5300	0.4960	0.4640	0.5260	0.7560	0.1940	0.2240	0.1760	0.2100	0.1860	0.2120	0.1980	0.3200	0.1820	0.1920	0.1200	0.1660	0.1980	0.1840	0.2360
0.5640	0.5280	0.4960	0.5560	0.6980	0.4720	0.2760	0.1980	0.2740	0.1740	0.2020	0.2060	0.3420	0.1360	0.2100	0.2240	0.1680	0.1960	0.2360	0.2420
0.5560	0.4920	0.5080	0.5800	0.7060	0.5300	0.6000	0.1940	0.2620	0.1180	0.2040	0.2100	0.3660	0.1620	0.2480	0.1480	0.2000	0.1980	0.2100	0.2680
0.5200	0.4160	0.4940	0.5100	0.6420	0.4260	0.5640	0.5380	0.2100	0.1640	0.1920	0.2000	0.3960	0.1340	0.2580	0.2520	0.2140	0.1980	0.1860	0.2380
0.5200	0.5080	0.5540	0.5540	0.6740	0.4760	0.5880	0.5460	0.5600	0.1840	0.2560	0.1860	0.3820	0.1480	0.1740	0.2120	0.1780	0.2000	0.2220	0.2380
0.5500	0.4600	0.5480	0.5280	0.6660	0.4560	0.5680	0.5620	0.5620	0.7180	0.1720	0.1900	0.3720	0.1580	0.2140	0.1800	0.2060	0.2320	0.2920	0.2140
0.6160	0.5160	0.4840	0.5540	0.6540	0.4740	0.6000	0.5520	0.5020	0.7000	0.7480	0.1800	0.3500	0.1440	0.1980	0.1660	0.2220	0.2000	0.1720	0.1960
0.5480	0.5140	0.4840	0.5800	0.6800	0.4580	0.5780	0.5480	0.5500	0.7020	0.7680	0.6100	0.3780	0.1160	0.1960	0.2200	0.2500	0.1760	0.1980	0.1760
0.6300	0.4940	0.5440	0.5780	0.6760	0.4960	0.5900	0.5600	0.4720	0.6900	0.7800	0.6120	0.7300	0.1300	0.1540	0.2180	0.2140	0.2000	0.2480	0.2160
0.5560	0.5380	0.5320	0.5620	0.6820	0.4800	0.6120	0.5360	0.4820	0.6680	0.7400	0.6040	0.6760	0.6840	0.1980	0.2740	0.1500	0.2140	0.2260	0.2020
0.5780	0.5680	0.5640	0.5860	0.7220	0.5280	0.6000	0.5480	0.5640	0.7160	0.7560	0.6620	0.7500	0.6560	0.7900	0.2340	0.1720	0.2040	0.2900	0.2000
0.5740	0.5040	0.5040	0.5720	0.6960	0.4840	0.5860	0.5660	0.5180	0.6220	0.7160	0.5740	0.6960	0.6460	0.7380	0.6700	0.1800	0.1840	0.2940	0.2300
0.5860	0.5420	0.5440	0.5940	0.7000	0.4700	0.6060	0.5440	0.5920	0.6540	0.7400	0.5880	0.7060	0.6440	0.7120	0.5900	0.7220	0.2100	0.2660	0.2120
0.5800	0.5080	0.5620	0.5960	0.7060	0.5020	0.6140	0.5760	0.5340	0.6820	0.7420	0.5720	0.7180	0.6140	0.7420	0.6220	0.6160	0.7200	0.2340	0.2160
0.5900	0.5440	0.5520	0.5780	0.6560	0.4960	0.6040	0.5560	0.5640	0.6820	0.7700	0.6140	0.6900	0.6040	0.7500	0.5960	0.6480	0.6280	0.6560	0.1620
0.5960	0.5620	0.5800	0.5800	0.7360	0.5280	0.6320	0.6160	0.6160	0.7300	0.7800	0.6300	0.7200	0.6320	0.7220	0.6260	0.6860	0.6100	0.7380	0.7540

```
Final Accuracy: 0.6537
Backward: 0.0149
Forward:  0.0075
```