ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

CVL Computer Vision Lab

Class-Incremental Learning for Tissue Classification

Semester Thesis

Ajaykumar Unagar
MSc Computational Science and Engineering
Department of Mathematics

Advisors: Prof. Dr. Orçun Göksel
Supervisor: Kevin Thandiackal

October 31, 2020

Abstract

Deep Neural Networks (DNNs) have been successfully applied to many problems in the medical domain. However, the common assumption of having access to the complete training data may not always hold true. Instead, data could be provided by different sources at different points in time. Especially in the medical field, privacy regulations can exacerbate this problem by prohibiting storage of old data. In such scenarios, proper incremental training for DNNs becomes essential to not forget previously learned knowledge. To this end, a recently published method called Deep Model Consolidation (DMC) proposes knowledge distillation with auxiliary data. We use this method as a benchmark and propose improvements on top of it. Furthermore, we present a novel approach to transform auxiliary images to look more like images from old datasets. With experiments on known benchmark datasets as well as a dataset for tissue type classification, we show that DMC and our image transformer-based method can reduce forgetting of previously acquired knowledge.

Acknowledgements

This work has been done in collaboration with IBM Research, Zurich. First, I thank Kevin for his continuous guidance throughout this project. Also, a huge shoutout to an amazing team at IBM, Dr. Maria Gabrani, Dr. Antonio Foncubierta, Pushpak Pati, and Guillaume Jaume for many fruitful discussions.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Image Classification is the problem of categorizing images into certain predefined classes. Deep Neural Networks (DNNs) are one of the most successful machine learning algorithms for this task. A common assumption while training any DNN is that the data from all image classes to be learned are available at the same time. Hence, network parameters are adapted to differentiate between all image categories for a given classification problem. However, this assumption is broken when all the classes for a given problem are not available at the same time and potentially arrive in different batches. In this incremental training scenario, when a neural network is trained on new data, previously adapted weights are overwritten. Hence, the neural network learns to classify new images, but *forgets* the classification task on previously trained data. This phenomenon known as *Catastrophic Forgetting* [21] is a fundamental limitation for DNN applications.

The ability to train DNNs incrementally is crucial, especially in the medical domain. An example problem is tissue type classification in microscopic images. DNNs are successfully applied in solving tissue type classification problems [12, 27, 10]. However, at training time, not all tissue categories are necessarily available. Often there are multiple providers for the tissue data and each one may only provide images of certain categories. If a DNN is trained only on currently available data, this will result in catastrophic forgetting of previous categories. An easy solution to overcome catastrophic forgetting would be to store the old data and keep retraining the network as the new data arrives along with the stored data. However, these datasets usually cannot be stored due to privacy concerns or legal obligations limiting usage of the data only to a certain project. Considering these restrictions, the successful deployment of DNNs in the medical domain requires more sophisticated methods that allow them to learn incrementally without forgetting. This underscores the importance of the research field of continual learning and in particular, class-incremental learning.

A large part of the research in continual learning follows these three directions: (1) replay a subset of samples from previous tasks to learn shared features, (2) isolate the unused parameters from the model and train new data only on those parameters, (3) regularize the network parameters to prevent large shifts in their values when moving to a new task. While these methods to a certain extent reduce forgetting in DNNs, the performance remains inferior to joint training on data of all classes.

Overall, replay-based methods are outperforming the other two types of methods [5]. However, since storing data in medical applications is usually prohibited due to privacy concerns, such methods cannot be used. Parameter-isolation methods work well in practice, but at the test time, usually require a task oracle that indicates to which task a given image belongs to. On the other hand, regularization-based approaches can work with a fixed network capacity and without using previous task data. Unlike parameter-isolation methods, regularization methods do not require a task-oracle at the test time and regularize the parame-

ters rather than freezing them. This helps in keeping the network size fixed and potentially only learning the features in addition to what already has been learned. Therefore, regularization-based methods are a compelling alternative for the class incremental learning in medical applications.

1.1 Focus of this Work

In this work we focus on Class-Incremental learning (CIL) for the tissue type classification problem. Class-Incremental learning is a much harder problem than task-incremental learning, since the network needs to differentiate between all learned class without any task oracle at test time. This setting is also known as single-head prediction in the literature [29]. Even though more difficult, the single-head setting is arguably more relevant in practice e.g., in the medical domain.

As for many types of medical data, privacy is also a major concern while working with microscopic images of human tissue. Since indefinitely storing the tissue data obtained from different providers is not feasible, we cannot rely on pure replay-based methods. Hence, in this work, we study data-focused regularization-based [5] methods. These methods use knowledge distillation [8] from the previous model while learning a new model with the new data. Knowledge distillation provides a learning signal for the previous tasks and ensures that the network does not completely forget the older tasks. We focus on the recently introduced Deep Model Consolidation (DMC) [32] method, which uses auxiliary data from a similar domain to consolidate knowledge from the old and the new model in a single model using distillation. This is particularly interesting for the tissue type classification, since tissue slices are very large, and only small parts of these slices are annotated for training. Hence, the remaining parts of these slices can be used as an auxiliary dataset for model consolidation. We study the limitations of this method when the task size is small (e.g., 2 or 3 classes per task). Furthermore, we analyze how the choice of the auxiliary data affects class-incremental learning on various datasets. Based on our findings, instead of directly using the auxiliary images, we propose to transform them so that they are more suitable for class-incremental learning.

1.2 Thesis Organization

In the chapter 2, we discuss prior work in the domain of continual learning, especially regularization-based methods. In chapter 3, we discuss DMC [32] and LwF [18] in detail and propose different modifications. We also propose a new method to transform auxiliary images for CIL. We present our results in chapter 4 and conclude in chapter 5

Chapter 2

Related Work

Incremental learning has been a long-standing research topic even before the popularity of Deep Neural Networks [28, 3]. McCloskey et al.[21] first identified *catastrophic forgetting*, where old memory is overwritten when a neural network is trained on new data. Similar forgetting effect in DNNs has been empirically shown by Goodfellow et al. [6]. Since then, the research in this field follows three directions:

2.1 Replay-Based Methods

Many replay-based methods have been shown to work well in incremental learning settings [24, 25, 4, 11] by storing samples from the previous tasks. In the medical domain, replay-based method had been used to incrementally build segmentation models for anatomical structures [22]. However, data-sensitive applications often do not allow users to store or in many cases even combine data from multiple providers. It is also possible to train generative models on the previous task data and use these models to replay samples while training classifier on the current dataset [2, 15, 23]. However, GAN training requires careful finetuning for the given dataset and is computationally expensive.

2.2 Parameter-Isolation Methods

Within constrained settings of the medical domain, parameter isolation methods with a fixed architecture [20, 19, 26] can also be used. These methods find less important parameters in the network and train only those parameters for the new data while keeping other parameters fixed. So, the current task uses previously learned important parameters, but do not update them. This approach allows parameter sharing between tasks, but they require a task oracle at test time to activate relevant parameters for a particular task. Aljundi et al. [1] proposed to learn multiple classifiers and an expert gate that at test time identifies the task an image belongs to. Once the task is identified, a designated classifier can be used to solve the task. However, since the number of classifiers grows with the number of tasks, this method is not scalable.

2.3 Regularization-Based Methods

Regularization-based methods are more suited for the class-incremental settings we are focusing on. Methods such as EWC [13], SI [31], and IMM [17] regularize the parameters of the network based on the priors from older tasks. Prior-based regularization ensures that, while training a network with the current data,

parameters do not deviate from the previous values by a large margin. However, such a soft penalty does not necessarily allow the network to remember previous tasks. On the other hand, in LwF [18], the authors proposed to use knowledge distillation of the previous tasks while fine-tuning classifier on the most recent data. Combining a cross-entropy loss of the current data with the distillation loss based on the output of previous classifier let the network to adapt current parameters for all the tasks. However, this method is highly biased towards the most recent task, because during training the authors only used current task data. In DMC [32], the authors attempted to alleviate this problem by using auxiliary data from a similar domain but potentially from a different distribution (to reduce forgetting of the older tasks). The authors train a separate network only on the most recent data, and then use the auxiliary data to consolidate knowledge of both the classifiers in a single model by knowledge distillation. Once a consolidated model is trained it can be used to classify all the classes of both the models.

It is important to mention that, combining multiple heterogeneous classifiers has been studied in other work [30] but in a different setting. The authors focus on a problem similar to DMC, but they assume that during consolidation, the model had access to all individual classifiers with the goal of combining them into one. Also, they used images from test data (of training data on which individual classifiers were trained) as an auxiliary data. Images from this test set come from the same distribution as the training set, that give the consolidation model more information than the images which are taken from a different distribution. Also, it is not practical to use test images for consolidation in CIL setting, since we do not have access to the test set of the current classes. Hence, this method is not suitable for CIL setting. Furthermore, while combining different models, the logits of the individual models might differ in scale. In [9], authors discussed the problem with imbalanced logits when combining multiple classifiers, and proposed cosine normalization techniques for the logits. However, this work has been tested in a setting with exemplar replay.

We use DMC [32] as a reference method since it works particularly well in CIL scenarios. Through extensive experiments we show the limitations of the current DMC approach and propose improvements by combining ideas from previous work [30, 9]. We also propose a new approach to use auxiliary data more effectively by learning useful transformations.

Chapter 3

Materials and Methods

As mentioned before, we focus on CIL for tissue type classification. In this domain, we cannot store samples from previous classes and therefore cannot employ pure replay-based methods. Moreover, at test time we do not have access to a task oracle to differentiate between tasks. Thus, we focus on distillation-based methods in a class-incremental setting, which is defined as follows.

3.1 Problem Definition

Assume that for an image classification problem, the image domain is $\mathcal{X}$ and we have N different categories such that $X_i \subset \mathcal{X}$ denotes the set of images belonging to class i. In CIL, we are given access to only a subset of these categories at a time. Assume that we have trained a model on the first *s* categories, and we receive new data with the images from categories *s*+1 to *t*. The goal of class-incremental learning is to further train the model without using the data of the previous *s* categories such that the classifier is able to accurately classify all t classes. In a practical scenario, data might arrive in many small batches of different classes, in which this method can easily be extended for more than 2 steps we presented here. It is crucial to notice that at test time we do not know which task a given image belongs to. Consequently, the network can classify all categories only if it learns discriminative features among all of them.

3.2 Methods

Within the CIL setting, we discuss Learning without Forgetting (LwF) [18] and DMC [32] to understand how knowledge distillation can be used to defy catastrophic forgetting in Deep Neural Networks.

3.2.1 LwF

This was the first work that demonstrated the effectiveness of knowledge distillation in a CIL setting. First, authors train a classifier C_1 on the previous data (fig. 3.1a), and a copy of this model is saved for the knowledge distillation. While tuning this classifier for the current data (called as C_2), authors remember the previous task using knowledge distillation loss (fig. 3.1b). Distillation loss forces the classifier C_2 to output the probabilities for the previous classes ($\hat{p}_1$) similar to the output probabilities of C_1 (p_1). Concurrently, the classifier C_2 also learns the current task via cross-entropy loss. Knowledge distillation from the previous model provides a learning signal to the current model such that the parameters are adapted to fit all the classes. However, notice that while training C_2, authors only used the data from the current task. Though,

the neurons for the previous classes are also forced to be activated by a distillation loss, while the data belongs to the current classes. This might create a conflicting signal for the classifier C_2. Also, C_2 is biased towards the current task because it has seen images only from this task.

3.2.2 DMC

DMC tries to overcome the bias in the training of LwF by using an auxiliary dataset $\mathcal{W}$ for distillation (fig. 3.3). Assume that we have a model C_1 trained on the previous *s* classes, and we get new data with images from classes *s*+1 to *t*. A classifier C_2 is trained from scratch only on the current data. Afterwards, both the classifiers (C_1 and C_2) are consolidated into a single classifier (C_{con}) by minimizing a double-distillation loss, defined for a single auxiliary sample as follows:

$$L_{dd}(y, \mathring{y}) = \frac{1}{t}\sum_{j=1}^{t}(y_j - \mathring{y}_j)^2. \tag{3.1}$$

Here, y_j denotes logits produced by the consolidated model, and $\mathring{y}$ is the concatenation of mean-normalized logit outputs of C_1 and C_2. Since these models have been trained individually, without any normalization, their logits can have very different values. The authors proposed the following mean normalization of logits:

$$\mathring{y}_j = \begin{cases} \hat{y}_j - \frac{1}{s}\sum_{i=1}^{s}\hat{y}_i, & 1 \le j \le s \\ \hat{y}_j - \frac{1}{t-s}\sum_{i=s+1}^{t}\hat{y}_i, & s < j \le t \end{cases}, \tag{3.2}$$

where $\hat{y}_j$ represents the logits produced by the original models. Shifting all logits by the mean does not affect the final probabilities of individual classifiers, but ensures that the logits in both the models have zero mean. However, even with the mean normalization, it is not guaranteed that both the models' logits will have the same scales. The consolidated model will still be biased towards the logits that have a larger scale.

As we can see in fig. 3.2, logits normalized by their mean still can have large scale differences. The logits for class 2 and 3 have median close to 0 compared to other class logits. One idea is to further normalize these logits or use other loss formulations to reduce the impact of logit scales. For balanced training of the consolidation model we propose the following normalization tricks:

3.2.3 Logit normalization tricks

To unify the scales of the logits from two different models, we applied the following normalization tricks.

(1) Variance normalization: One simple way to make sure that the logits from both the models have the same scale is to apply variance normalization in addition to mean normalization as in eq 3.3.

$$\mathring{y}_j^{norm} = \begin{cases} \frac{\mathring{y}_j}{Var_{i=1}^{i=s}(\mathring{y}_i)}, & 1 \le j \le s \\ \frac{\mathring{y}_j}{Var_{i=s+1}^{i=t}(\mathring{y}_i)}, & s < j \le t \end{cases} \tag{3.3}$$

(2) Cross-Entropy normalization : Similar to the previous work [30], we also use the probability normalization. To implement this, we first need to convert the logits of the individual models and the consolidated model into probabilities by applying a softmax layer. For the individual models, we apply softmax activation to the respective model outputs independently and get the target probability $\mathring{p}_j$ for each

class. For the consolidation model, we divide the logits into two parts corresponding to the output classes of two individual models as follows:

$$p_j = \begin{cases} \frac{e^{y_j}}{\sum_{i=1}^{s} e^{y_i}}, & 1 \leq j \leq s \\ \frac{e^{y_j}}{\sum_{i=s+1}^{t} e^{y_i}}, & s < j \leq t \end{cases}, \tag{3.4}$$

and can then apply the cross-entropy loss:

$$L_{dd}(y, \mathring{y}) = \sum_{j=1}^{t} -\mathring{p}_j * \log p_j. \tag{3.5}$$

(3) Cosine normalization : Instead of applying normalization post computation of the logits, authors in [9] proposed to generate normalized logits. Suppose, the final layer input features are $f(x)$ and the weights are θ and bias b, and logits are defined as

$$y_j = \theta_j^T f(x) + b_j. \tag{3.6}$$

Without normalization, scale of the individual logits depends a lot on the scale of the weight vector θ_j, the scale of the embedding $f(x)$, and bias values for each output neuron. We can generate normalized individual logits by normalizing weight and embedding vectors as follows:

$$y_j = \bar{\theta}_j^T \bar{f}(x), \tag{3.7}$$

where, $\bar{v} = \frac{v}{||v||_2}$, denotes a normalized vector.

Note that in eq. 3.7, we do not use bias term unlike in eq. 3.6. In addition, we use the dot product of the normalized weights and the embedding which ensures that all logits are within the range -1 to 1.

3.2.4 Image Transformer Networks

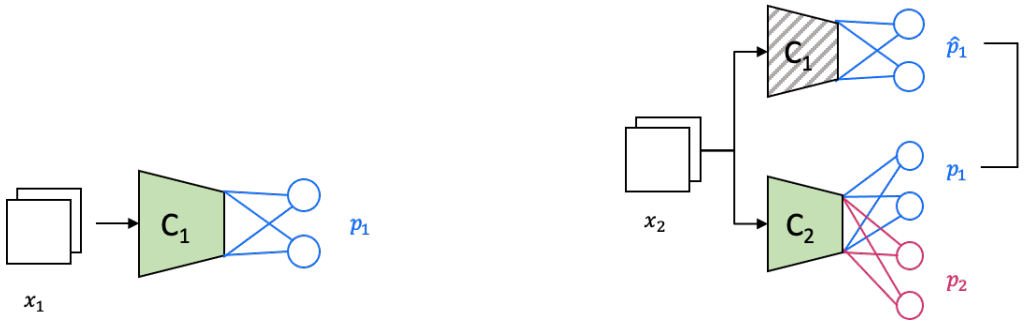
Another limitation of the DMC method is that two models are trained independently on the images from different classes. After that, the consolidated model is trained using auxiliary data and it never observes the images for the classes it needs to classify. This training approach might not learn discriminative features useful for classification among all the classes. Hence, instead of using two separate classifiers, we propose to use a single classifier similar to LwF [18]. Moreover, in DMC authors used the auxiliary data directly and they did not have control over bias in the auxiliary data towards certain classes. We propose to transform the auxiliary images such that they are more representative of the previous image categories, instead of using them directly for the distillation. A schema for this method is shown in Fig. 3.4. We define this method for the two steps of incremental training, but can be extended for more than two steps.

We train a classifier C_1 on the first task. After the classifier training, we train a transformer T_1 on auxiliary data x_{aux} such that the output x'_{aux} of the transformer, when passed through the classifier C_1, matches the output of the actual task 1 data x_1. We use a moment matching loss on individual output probabilities to train the transformer T_1. We try to reduce the L2 distance between the mean and the variance vectors $\mu_{1,aux}$, $\sigma_{1,aux}$ and μ_1, σ_1, which is defined in eq. 3.8 This transformed auxiliary data x'_{aux} should be more similar to x_1 than x_{aux} to x_1. After transformer training, we expand C_1 to accommodate additional image categories of the current dataset. This model C_2 is initialized with the weights of C_1 and during training, the transformed auxiliary data is used for the knowledge distillation of C_1. Since we are using only

a single classifier all the time, the classifier C_2 should learn to classify all image categories seen until now, as long as $x_{aux}{}'$ is sufficiently similar to the actual x_1.

$$L_{moment} = (\mu_1 - \mu_{1,aux})^2 + (\sigma_1 - \sigma_{1,aux})^2 \tag{3.8}$$

where μ_1 and σ_1 are obtained on full data x_1, while $\mu_{1,aux}$ and $\sigma_{1,aux}$ are statistics of a single batch of the auxiliary data.



(a) (b)

Figure 3.1: Learning without forgetting (LwF). The first classifier is trained using method (a). All subsequent classifiers are trained using method (b). After training of (b) C_1 is replaced by C_2 for the next step.

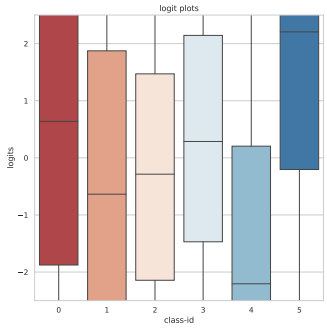
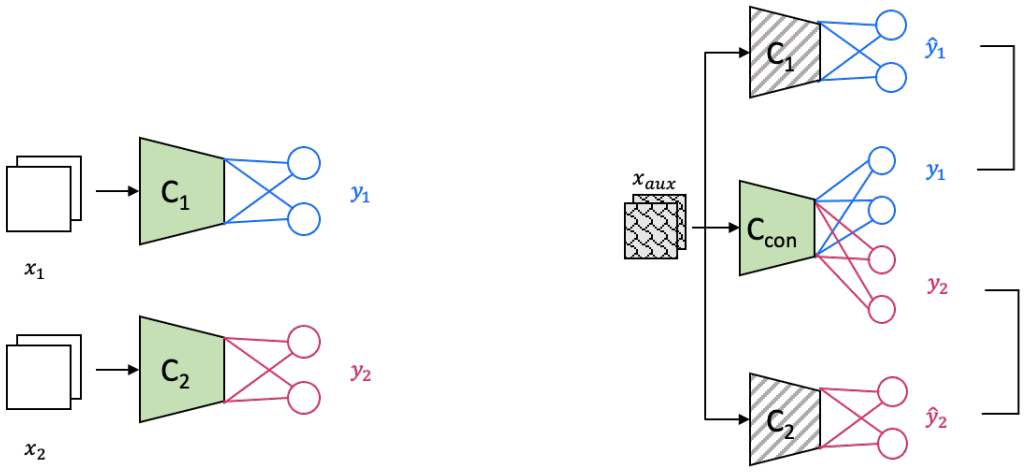


Figure 3.2: Model logits for individual models. Class 0, 1, 2, 3 belongs to C_1 and 4, and 5 belongs to C_2



(a) (b)

Figure 3.3: Deep Model Consolidation (DMC) method has two steps. On the left two classifiers are trained independently on different class of images. On the right, a consolidated model is trained using auxiliary data and double distillation. Green = trainable modules, striped = frozen modules

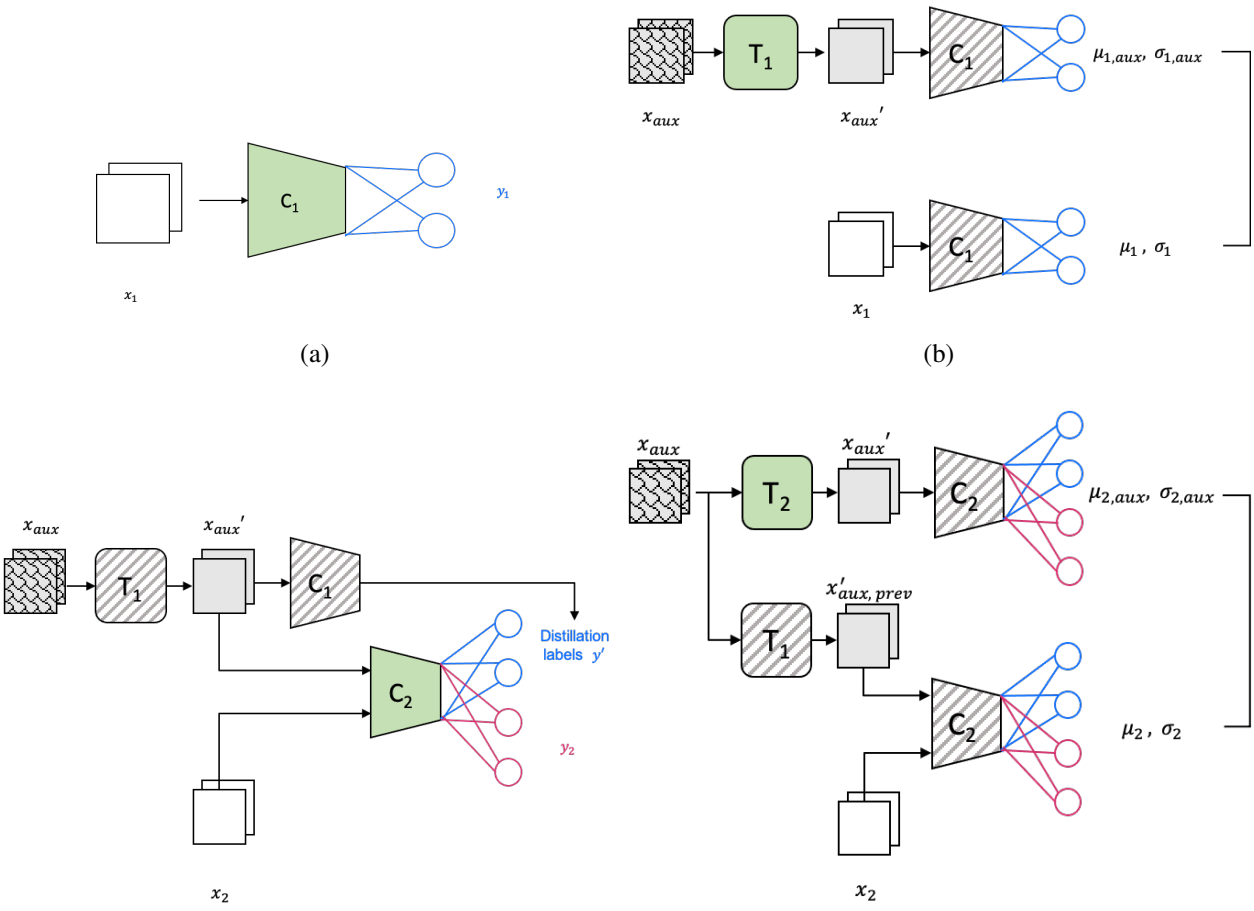


Figure 3.4: Knowledge Distillation with transformed auxiliary data. (a) represents first classifier training, while (c) represents all subsequent classifiers training. Similary, (b) represents first transformer training and (d) all subsequent transformers training. In (b) and (d) we use L2-loss to match moments of output probabilities. green = trainable modules, striped = frozen modules.

Chapter 4

Experiments and Results

4.1 Datasets

In this work, we specifically focus on evaluating CIL in a small task setting, which is more challenging. Networks trained to classify a small number of image categories may only learn features that are sufficient to distinguish these categories, and they might not generalize to future tasks. Furthermore, small tasks provide more flexibility for medical applications, since each data provider is not required to annotate a large number of different classes. We evaluate the different methods described in Section 3 on the following three datasets.

(1) Split-MNIST: The MNIST [16] dataset has 60,000 training images and 10,000 test images of 28x28 pixels. We divide the training set into two parts. The images from class 0 to 5 are used for incremental training and the remaining 4 categories (6 to 9) serve as auxiliary data. We further divide the training images into three tasks with two classes each.

(2) Split-CIFAR10: CIFAR-10 [14] consists of 60,000 images of size 32x32 pixels from 10 different categories. Out of these, 50,000 images are part of the training set and 10,000 images are part of the test set. Similar to Split-MNIST, we use images from the first 6 classes for the CIL setting of three tasks, and images from the remaining 4 categories as auxiliary data.

(3) Tissue Classification: We use the Tissue Classification data from [12] that comprises 100,000 non-overlapping image patches from hematoxylin & eosin (H&E) stained histological images of human colorectal cancer (CRC) and normal tissues. The dataset has images for 9 tissue types. We divide the 6 categories debris (DEB), cancer-associated stroma (STR), background (BACK), lymphocytes (LYM), adipose (ADI), and mucus (MUC) into three tasks with two categories each. The images from the remaining 3 tissue types, i.e., smooth muscle (MUS), normal colon mucosa (NORM), and colorectal adenocarcinoma epithelium (TUM), are used as auxiliary data.

4.2 Network Architecture and Training details

We design different architectures for each dataset. These architecture have been chosen based on complexity of the dataset.

For **Split-MNIST**, we use a simple LeNet architecture with 3 convolution layers of filter sizes 5x5 with 6, 16, and 120 filters per layer, respectively. After the first and the second convolution layer, we add a max-pool layer of size 2x2, and after the last convolution layer, we add a dense layer with 84 neurons. The final

layer has as many output neurons as there are classes and is followed by a softmax layer that converts them into probabilities. We use ReLU activations everywhere except for the final layer.

For **Split-CIFAR10**, we use three convolutional blocks followed by a dense layer. Each block has 2 convolution layers with filter-size 3x3. The first layer has stride 1 and preserves the input size. The second layer has stride 2 and reduces the input size by half. Each block has 64, 128, and 256 channels, respectively. The flattened output of the last convolutional block is fed to a dense layer with 128 hidden units, followed by an output layer with the number of neurons being equal to the number of output classes. All layers have ReLU activations, except for the output layer where we use the standard softmax activation.

For **Tissue Classification** dataset we use a pretrained ResNet-18 [7] feature extractor followed by the output layer with the number of neurons being equal to the number of output classes.

For DMC, while doing consolidation, we use the same model architecture as the individual classifiers and we start the consolidated model training from scratch. For the transformer, we use three convolution layers with filter size 3x3. The first and second layer have 16 and 32 channels, respectively. The final layer has the same number of channels as the input image. However, without any kind of regularization, the transformers can considerably modify the images such that they no longer look realistic. Hence, we use L2-loss between transformed and an original image as a regularizer. This regularizer allows transformer to learn statistics of the original model without shifting the images by a large margin.

For all the experiments we use stochastic gradient descent (SGD) with 0.01 learning rate, 0.9 momentum and 0.0005 weight decay, for 50 epochs. We use a batch-size of 128 for all the datasets except for the transformer training. For transformers training, we use a batch-size of 2048 for Split-MNIST and Split-CIFAR-10, and 256 for the Tissue Classification data (due to memory constraints).

4.3 Results

We compare different variants of DMC on the small tasks with our transformer-based method on three datasets. Results for these variants are shown in Table 4.1, 4.2, and 4.3. In these tables, each column represents for how many classes a classifier is trained on. Each row represents the accuracy of a CIL variant on the test data after each incremental training step. The method *finetune* represents incremental training of the same classifier only using the current data. This should be the lower bound for any CIL method. On the other hand, *upper bound* is obtained by training a classifier jointly on the classes up to and including the ones mentioned in corresponding columns. The best CIL variant is marked in **blue**.

The images in the Split-MNIST dataset are rather simpler since they have only one channel. In this setting, we can see that our transformer-based method outperforms DMC for three steps of incremental training (Table 4.1). As we have discussed in Chapter 3, the accuracy of DMC for a particular task depends on whether the auxiliary data is unbiased. In Table 4.4, we see that DMC suffers from severe forgetting for task-2 compared to task-1 and task-3. This can also be explained by the logit scale difference we shown in fig. 3.2. However, by transforming the auxiliary data, we observe a graceful forgetting for these tasks. In addition, the final accuracy of the classifier is better than DMC. In fig. 4.1 we show transformed auxiliary images for Split-MNIST after the first transformer training. The labels on top of the images represent classifier output which is trained to classify images as 0 or 1. These images are visually similar to actual auxiliary images, because of the regularization loss. But, we can see that the digits with curved edges are more likely to be classified as 0 than the digits with the straight edges, which are classified as 1. This suggests that the transformer activated some parts of the images more than the others, and hence our transformed data is better for the distillation of the previous tasks than the original auxiliary data.

For Split-CIFAR10 and the Tissue Classification data, the content of the images varies considerably

Figure 4.1: Transformed auxiliary images for Split-MNIST dataset

from one class to the other. In this setting, transformer-based methods are not on par with DMC. One reason could be that the transformed images are not representative of the previous tasks. Since the transformers are trained using a moment matching loss, there is no signal to learn if the transformed images are actually similar to the training images. In additional, the moment matching loss only matches the output probabilities, leaving many degrees of freedom for transformations in the input space. Transformed auxiliary images are shown in fig. 4.2 after the first transformer training. The labels on top of the images represents output of the classifier trained to classify images as aeroplane or automobile. Since there is less feature overlap between training images and auxiliary images, we can not observe any pattern for the transformed images unlike for split-MNIST.

We see that for all the datasets, DMC with variance normalization is comparable to the original DMC accuracy. However, cosine and cross-entropy normalization decrease the accuracy of the classifier in all three datasets. This can be explained by the fact that these normalization techniques have been proposed for different settings than CIL. For example cross-entropy normalization in eq. 3.4 assumes that the probabilities of the consolidated model add up to 2 because we apply two individual softmax to the final layer. On the other hand, in cosine normalization logits for both the model (C_1 and C_2) are between range -1 to 1. It is possible that auxiliary images activate logits for both the models and they might create conflicting signals for the consolidated model.

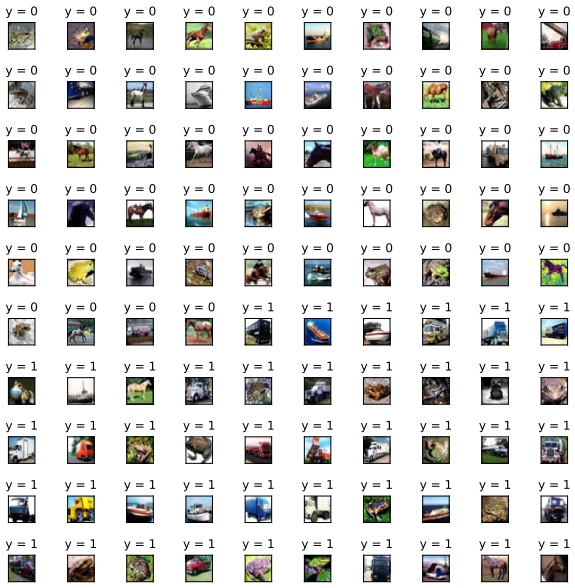


Figure 4.2: Transformed auxiliary images for Split-CIFAR dataset

Method	Number of classes		
	2	4	6
DMC	0.999	**0.943**	0.733
DMC+var norm	0.999	0.941	0.716
DMC+cross-entropy	0.999	0.753	0.162
DMC+cosine norm	0.999	0.771	0.426
Transformer	0.999	0.900	**0.833**
Finetune	0.999	0.490	0.311
Upper bound	0.999	0.997	0.993

Table 4.1: Split-MNIST

Method	Number of classes		
	2	4	6
DMC	0.957	0.593	0.461
DMC+var norm	0.957	0.598	**0.466**
DMC+cross-entropy	0.957	0.393	0.167
DMC+cosine norm	0.957	0.568	0.408
Transformer	0.957	**0.6435**	0.404
Finetune	0.957	0.432	0.308
Upper bound	0.957	0.849	0.806

Table 4.2: Split-CIFAR-10

Method	Number of classes		
	2	4	6
DMC	0.976	0.809	**0.712**
DMC + var norm	0.976	**0.822**	0.710
DMC + cross-entropy	0.976	0.484	0.426
DMC + cosine norm	0.976	0.273	0.607
Transformer	0.976	0.371	0.461
Finetune	0.976	0.661	0.513
Upper bound	0.976	0.965	0.974

Table 4.3: Tissue type Classification

	Train		
Test	task-0	task-1	task-2
task-0	0.999	0.964	0.904
task-1	-	0.921	0.400
task-2	-	-	0.901

	Train		
Test	task-0	task-1	task-2
task-0	0.999	0.806	0.742
task-1	-	0.998	0.779
task-2	-	-	0.994

Table 4.4: Task level accuracy for DMC (left) and Transformer based method (right) on Split-MNIST

Chapter 5

Conclusion

Distillation-based regularization approaches provide a good alternative to other CIL methods. These approaches can use auxiliary data from a different distribution than the training data, which gives us more flexibility to apply these methods, e.g., in the medical domain. Furthermore, distillation does not require labels for the auxiliary data, that are much harder to obtain in the medical domain (compared to natural images). However, we observed that the auxiliary data can be biased towards certain classes of the training set and DMC does not present a way to choose unbiased auxiliary images. We proposed to use auxiliary data combined with a transformation model. This model learns to transform the auxiliary images more similar to the previous task images. We showed that such a transformation can remove the bias towards certain classes in the auxiliary data. Unbiased auxiliary images show a graceful forgetting of the previous tasks which is more predictable than DMC.

We see that transformer training is not effective for the images with a complex structure, since these images give a transformer many degrees of freedom to modify them. Even though these images are visually similar to the original auxiliary images, we believe that for the classifier, the images produced by the transformer are too different from the previous task images. One possible argument for this deteriorating behavior is that the transformer needs to learn a better objective than moment matching. Generative Adversarial Networks (GANs) allow us to generate images from a specific distribution post-training. However, most of the approaches until now train GAN to produce images from a random noise vector. As the next steps, one can explore GAN and other image transformation architectures.

Bibliography

[1] Rahaf Aljundi, Punarjay Chakravarty, and Tinne Tuytelaars. Expert gate: Lifelong learning with a network of experts. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 3366–3375, 2017.

[2] Craig Atkinson, Brendan McCane, Lech Szymanski, and Anthony Robins. Pseudo-recursal: Solving the catastrophic forgetting problem in deep neural networks. arXiv preprint arXiv:1802.03875, 2018.

[3] Gert Cauwenberghs and Tomaso Poggio. Incremental and decremental support vector machine learning. In Advances in neural information processing systems, pages 409–415, 2001.

[4] Arslan Chaudhry, Marcus Rohrbach, Mohamed Elhoseiny, Thalaiyasingam Ajanthan, Puneet K Dokania, Philip HS Torr, and M Ranzato. Continual learning with tiny episodic memories. 2019.

[5] Matthias De Lange, Rahaf Aljundi, Marc Masana, Sarah Parisot, Xu Jia, Ales Leonardis, Gregory Slabaugh, and Tinne Tuytelaars. Continual learning: A comparative study on how to defy forgetting in classification tasks. arXiv preprint arXiv:1909.08383, 2(6), 2019.

[6] Ian J Goodfellow, Mehdi Mirza, Da Xiao, Aaron Courville, and Yoshua Bengio. An empirical investigation of catastrophic forgetting in gradient-based neural networks. arXiv preprint arXiv:1312.6211, 2013.

[7] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 770–778, 2016.

[8] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. arXiv preprint arXiv:1503.02531, 2015.

[9] Saihui Hou, Xinyu Pan, Chen Change Loy, Zilei Wang, and Dahua Lin. Learning a unified classifier incrementally via rebalancing. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 831–839, 2019.

[10] Osamu Iizuka, Fahdi Kanavati, Kei Kato, Michael Rambeau, Koji Arihiro, and Masayuki Tsuneki. Deep learning models for histopathological classification of gastric and colonic epithelial tumours. Scientific Reports, 10(1):1–11, 2020.

[11] David Isele and Akansel Cosgun. Selective experience replay for lifelong learning. arXiv preprint arXiv:1802.10269, 2018.

[12] Jakob Nikolas Kather, Johannes Krisam, Pornpimol Charoentong, Tom Luedde, Esther Herpel, Cleo-Aron Weis, Timo Gaiser, Alexander Marx, Nektarios A Valous, Dyke Ferber, et al. Predicting survival from colorectal cancer histology slides using deep learning: A retrospective multicenter study. PLoS medicine, 16(1), 2019.

[13] James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, et al. Overcoming catastrophic forgetting in neural networks. Proceedings of the national academy of sciences, 114(13):3521–3526, 2017.

[14] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.

[15] Frantzeska Lavda, Jason Ramapuram, Magda Gregorova, and Alexandros Kalousis. Continual classification learning using generative models. arXiv preprint arXiv:1810.10612, 2018.

[16] Yann LeCun. The mnist database of handwritten digits. http://yann. lecun. com/exdb/mnist/, 1998.

[17] 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, pages 4652–4662, 2017.

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

[19] Arun Mallya, Dillon Davis, and Svetlana Lazebnik. Piggyback: Adapting a single network to multiple tasks by learning to mask weights. In Proceedings of the European Conference on Computer Vision (ECCV), pages 67–82, 2018.

[20] Arun Mallya and Svetlana Lazebnik. Packnet: Adding multiple tasks to a single network by iterative pruning. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 7765–7773, 2018.

[21] Michael McCloskey and Neal J. Cohen. Catastrophic interference in connectionist networks: The sequential learning problem. Academic Press, Psychology of Learning and Motivation, 24:109 – 165, 1989.

[22] Firat Ozdemir and Orcun Goksel. Extending pretrained segmentation networks with additional anatomical structures. International journal of computer assisted radiology and surgery, 14(7):1187–1195, 2019.

[23] Jason Ramapuram, Magda Gregorova, and Alexandros Kalousis. Lifelong generative modeling. arXiv preprint arXiv:1705.09847, 2017.

[24] Sylvestre-Alvise Rebuffi, Alexander Kolesnikov, Georg Sperl, and Christoph H Lampert. icarl: Incremental classifier and representation learning. In Proceedings of the IEEE conference on Computer Vision and Pattern Recognition, pages 2001–2010, 2017.

[25] David Rolnick, Arun Ahuja, Jonathan Schwarz, Timothy Lillicrap, and Gregory Wayne. Experience replay for continual learning. In Advances in Neural Information Processing Systems, pages 350–360, 2019.

[26] Joan Serrà, Didac Suris, Marius Miron, and Alexandros Karatzoglou. Overcoming catastrophic forgetting with hard attention to the task. arXiv preprint arXiv:1801.01423, 2018.

[27] Chetan L Srinidhi, Ozan Ciga, and Anne L Martel. Deep neural network models for computational histopathology: A survey. Medical Image Analysis, page 101813, 2020.

[28] Nadeem Ahmed Syed, Syed Huan, Liu Kah, and Kay Sung. Incremental learning with support vector machines. 1999.

[29] Gido M van de Ven and Andreas S Tolias. Three scenarios for continual learning. arXiv preprint arXiv:1904.07734, 2019.

[30] Jayakorn Vongkulbhisal, Phongtharin Vinayavekhin, and Marco Visentini-Scarzanella. Unifying heterogeneous classifiers with distillation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 3175–3184, 2019.

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

[32] Junting Zhang, Jie Zhang, Shalini Ghosh, Dawei Li, Serafettin Tasci, Larry Heck, Heming Zhang, and C-C Jay Kuo. Class-incremental learning via deep model consolidation. In The IEEE Winter Conference on Applications of Computer Vision, pages 1131–1140, 2020.

Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Declaration of originality

The signed declaration of originality is a component of every semester paper, Bachelor's thesis, Master's thesis and any other degree paper undertaken during the course of studies, including the respective electronic versions.

Lecturers may also require a declaration of originality for other written papers compiled for their courses.

I hereby confirm that I am the sole author of the written work here enclosed and that I have compiled it in my own words. Parts excepted are corrections of form and content by the supervisor.

Title of work (in block letters):

CLASS-INCREMENTAL LEARNING FOR TISSUE CLASSIFICATION

Authored by (in block letters):

For papers written by groups the names of all authors are required.

Name(s):	**First name(s):**
Unagar	Ajaykumar

With my signature I confirm that

- I have committed none of the forms of plagiarism described in the 'Citation etiquette' information sheet.
- I have documented all methods, data and processes truthfully.
- I have not manipulated any data.
- I have mentioned all persons who were significant facilitators of the work.

I am aware that the work may be screened electronically for plagiarism.

Place, date	**Signature(s)**
Zurich / 31st Oct, 2020	

For papers written by groups the names of all authors are required. Their signatures collectively guarantee the entire content of the written paper.