CONVAER: CONVOLUTIONAL VARIATIONAL AUTOENCODERS FOR INCREMENTAL SIMILARITY LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Due to catastrophic forgetting, incremental similarity learning in neural networks remains an open challenge. Previous work has shown that keeping *image exemplars* during incremental similarity learning is effective as a proxy representation for base knowledge (past learned features and embeddings). It is also generally accepted that the output layers learn more task-specific feature embeddings during the later training stages compared to the input layers' general features earlier on. Building on these insights, we start by freezing the input layers of a neural network. We then investigate the viability of generating "embedding" exemplars from a VAE that can protect base knowledge in the intermediate to output layers of the neural networks. These generated exemplars replace the necessity for retaining images from previously learned classes. We experimented with three metric learning loss functions on the CUB-200 and CARS-196 in an incremental similarity learning setup. We train different VAEs to generate exemplars from the intermediate convolution layers and linear output layers. We use these generated exemplars to represent base knowledge. We compared our work to a previous technique that stores image exemplars. The comparison is done for base knowledge, new knowledge, and average knowledge preservation as metrics.

The state-of-the-art results show that generating exemplars from the linear and convolutional layers retained the highest ratio of base knowledge. We note that using embeddings from the linear layers leads to better performance on new knowledge than convolutional embeddings. Overall our methods yield better average knowledge performance across all experiments. These results support the view that for incremental similarity learning to overcome catastrophic forgetting, emphasis can be placed on learning embedding exemplars for intermediate to output layers. Further, we note that most incremental similarity learning for new classes depends on the linear layers rather than the convolutions. Further investigation into the relationship between transfer learning and similarity learning and the protection of intermediate layer embedding space for catastrophic forgetting is required.

1 INTRODUCTION

In machine learning, incremental learning is the process of updating a model as new data becomes available or extended to support further tasks. An incrementally trained model should ideally retain previously attained knowledge while incorporating any new knowledge made available as it trains Syed et al. (1999); Polikar et al. (2001). Many machine learning algorithms cannot retain prior knowledge or do so in an unsatisfactory manner. Models that do not incrementally learn new tasks whilst retaining prior knowledge suffer from catastrophic forgetting. Catastrophic forgetting typically occurs during training on new data that contains no or highly imbalanced examples drawn from prior learned distributions McCloskey & Cohen (1989); Ratcliff (1990).

Catastrophic forgetting in deep neural networks and virtually all of the tasks supported by them remains an open research problem Goodfellow et al. (2013); Fernando et al. (2017); Robins (1995); Draelos et al. (2017). Historically, analyses have been focused almost entirely on incremental supervised classification in multi-layer perceptrons (MLP) neural networks such as typically encountered in computer vision tasks. However, there persists a lack of evidence detailing the degree to which similarity learning tasks and the underpinning pair mining and loss functions are affected by catastrophic forgetting. Further, most techniques aimed at overcoming catastrophic forgetting in deep neural networks have been engineered with classification tasks in mind Rannen et al. (2017); Rebuffi et al. (2017); Choi et al. (2019).

Previous work by Huo & van Zyl (2021) compared four state-of-the-art algorithms for reducing catastrophic forgetting during incremental learning. These algorithms included Fully Connected VAEs (FullVAE), Elastic Weight Consolidation Kirkpatrick et al. (2017), Encoder-Based lifelong learning Rannen et al. (2017), and incremental Classifier and Representation Learning iCaRL Rebuffi et al. (2017). Their work analyzed several loss functions on MNIST, EMNIST, Fashion-MNIST, and CIFAR-10. All the considered techniques were effective but to an unsatisfactory extent with FullVAE and iCaRL shown to be the most robust across numerous loss functions. However, given that the datasets used are somewhat trivial, these results only partially indicate what is to be expected on real-world data.

This paper builds on this prior body of research and presents conclusions on the influence of catastrophic forgetting in incremental metric learning. We approximated the catastrophic forgetting test procedure of Kemker et al. (2018) which is described for classification tasks. Our work examines three-loss functions, contrastive, centre and triplet loss, using CUB200 Welinder et al. (2010) and CAR196 Krause et al. (2013) in metric learning. We contrast the current state of the art for catastrophic forgetting during incremental metric learning, FullVAE and iCaRL, to our solution ConVAEr. ConVAEr uses convolutional Variational Autoencoders (VAE) to generate representations fed into the convolutional layers to supplement previously seen knowledge without regenerating entire images or the need to keep a collection of previous data. We present results that allow the reader to explore which technique retains the most base knowledge, new knowledge, and overall knowledge during incremental metric learning.

1. We show that our method yields better average knowledge retention across all experiments.
2. We support the importance of keeping prior knowledge or data during incremental similarity learning.
3. We demonstrate that injected VAE generated representations work as well as images exemplars.
4. We show that intercepting exemplars from the convolutional layers retained the highest ratio of base knowledge.
5. We note that using embeddings from the linear layers leads to better performance on new knowledge than convolutional embeddings.

2 Related Work

2.1 Catastrophic forgetting

Goodfellow et al. (2013) investigated catastrophic forgetting in gradient-based neural networks used for classification. The results showed that various combinations of activation function and learning were affected differently by catastrophic forgetting. The work by Rannen et al. (2017) demonstrated the problem of catastrophic forgetting in deep convolutional neural networks (DNN) AlexNet. They highlighted the classification performance dropped in a previously learned task when a DNN is fined-tuned for newer classification tasks. The authors proposed using lightweight autoencoders to “store” the embedding learned by the base network. A new autoencoder is trained and kept after the network learns each new task. The method significantly reduced catastrophic forgetting during incremental classification.

Choi et al. (2019) proposed the use of an autoencoder-based incremental metric learning method for classification without the use of a softmax classification layer. The work is premised on the notion of a metric-based classification method, nearest-class-mean (NCM)Mensink et al. (2013). A pre-trained fixed network was used as a feature extractor and the autoencoder is trained on the feature embeddings. To overcome catastrophic forgetting while fine-tuning the autoencoder to new classes, the authors use regularization techniques: Synaptic Intelligence (SI) Zenke et al. (2017), and

Memory Aware Synapses (MAS) Aljundi et al. (2018). The methods demonstrated good memory retention without the need to train on older data. Our work used VAE for knowledge preservation to supplement the update of the network during incremental similarity learning whereas the authors used a fixed network to supply the autoencoder for incremental classification and not similarity learning.

2.2 Incremental Classifier and Representation Learning (iCaRL)

Incremental Classifier and Representation Learning (iCaRL) is a method proposed by Rebuffi et al. (2017) for reducing catastrophic forgetting. It is reported to learn classes incrementally over a longer period than other methods. iCaRL relies primarily on storing exemplars of previously seen classes. Each class's exemplar set is constructed, using the herding algorithm, from the k closest images to the class's mean representation. The stored exemplars are used to supplement the incremental learning phase of new classes using knowledge distillation. Since iCaRL does not directly translate to similarity learning tasks we implemented the modified version as described in the paper by Huo & van Zyl (2021).

2.3 Encoder-Based Lifelong Learning

Rannen et al. (2017) proposed Encoder-Based Lifelong Learning (EBLL) for incremental learning in classification tasks. The method modifies how the convolutional layers of a network are updated. After each incremental learning task, a single autoencoder is trained to reconstruct the "images" at the output of the convolutional layers. The reconstructed images are passed through the network's remaining fully connected layers to calculate their resulting classification loss. The reconstruction loss, together with the classification loss, is used to update the autoencoder.

The previous task's classification layer (output layer) is detached for each new incremental learning task, and a new classification layer is attached. A frozen copy of the previous optimal network is made before training the next incremental task. The new images are passed through both the new and frozen network during training for the new task to calculate the distillation loss and added to the new classification loss and the encode loss to update the new network Rannen et al. (2017). However, only the new network is updated. For updating the network's weights, the images' convolutional layers outputs of the new and frozen network are passed into the autoencoder up to the bottleneck layer, where the mean square error is calculated and added to the classification loss and propagated through the autoencoders' network's weights. This process constrains the weight update of the convolutional network layers to compromise between new and old tasks.

2.4 Fully Connected VAE

Huo & van Zyl (2021) proposed using a VAE to represent images in a representation that can be passed through intermediate layers in the network. The authors focused on preserving knowledge from the fully connected layers (flattened output from the last CNN layer). Their method was shown to outperform iCaRL and other methods using similar test metrics to Kemker et al. (2018) during incremental similarity learning. The concern with their work was that they experimented using a simple network and a variety of simple datasets that are not representative of real-world performance.

2.5 Our approach (ConVAEr)

Rannen et al. (2017) constrain the weights of the feature extraction layers (convolutions) that were optimal for previous tasks with an autoencoder. The solution is robust when reusing the feature extraction layers (convolutions) on new tasks. Each task is tested independently from the others with its classification layer—the approach yields encouraging results by finding a middle-ground across the base and new knowledge. Huo & van Zyl (2021), however, have previously demonstrated that this approach is not practical for incremental metric learning.

The iCaRL method by Rebuffi et al. (2017) depends on the storage of a large number of exemplars. As reported, the performance of iCaRL decreases with time as the number of exemplars per class is reduced to accommodate new classes. Eventually, the stored exemplars are not sufficient to represent all classes. Further, retaining previous exemplars is hardly akin to overcoming catastrophic forgetting. Instead, this represents retaining previous data rather than previously learned knowledge. However, iCARL does present as a state-of-the-art method and, as such, provides a reasonable baseline for comparison.

Huo & van Zyl (2021) combine ideas from Rannen et al. (2017) and Rebuffi et al. (2017). They train a new variational autoencoder (VAE) for each class. The VAEs learn representations at the end of the convolutional layers. The use of VAEs allows them to generate examples from previously seen classes. The method requires that the convolutional layers be frozen after initial training or pre-trained frozen convolutional layers from a base model are used. The VAEs generate representations from previously seen classes combined with the new classes during incremental metric training to perform incremental metric learning. The work of Huo & van Zyl (2021) more closely aligned with the intent of overcoming catastrophic forgetting. That is, using previously learned representations and knowledge for future metric learning tasks. We extend on the work by Huo & van Zyl (2021) by using the authors method and moving the VAE to intercept input to the convolution layer instead of at the input to the fully connected layers show in figure 2.

The autoencoder's reconstruction loss function is required to vary depending on the network's interception layer's activation function. For example, in our case, the convolutional interception layers use ReLU activation, and consequently, we used the Binary Cross-Entropy objective function to determine the reconstruction errors summed with the Kullback-Leibler divergence. The loss function to update the VAEs is given as:

$$\begin{aligned} L_{VAE} = -\frac{1}{N}\sum_{i=1}^{N} y_i \cdot \log(p(y_i)) + (1-y_i) \cdot \log(1-p(y_i)) \\ +\frac{1}{2}(\exp(\sigma^2) + \mu^2 - 1 - \sigma^2), \end{aligned} \tag{1}$$

where σ^2 is the variance of the entire dataset and μ is the mean. The first term is the Binary Cross-Entropy reconstruction loss, and the second term is the Kullback–Leibler divergence.

We also make use of the angle-wise distillation loss seen the paper by Huo & van Zyl (2021) on the examples generated from the VAEs. This usage with the network update during incremental learning, is similar to the approach taken by iCaRL Rebuffi et al. (2017).

The incremental learning step makes use of the following loss function defined as:

$$L_{\text{incremental}} = l_{\text{metric learning}} + \lambda_{\text{distil}} * L_A \tag{2}$$

, where $l_{\text{metric learning}}$ are the three possible metric learning functions, we state below. L_A is the angle-wise distillation for metric learning. λ_{distil} is the importance that we place on the angle-wise distillation loss. The student output is the convolution image output we get from the network incrementally training at each new train step. The teacher output is the convolution output image we get from the frozen network before performing the new incremental train step.

For center loss, we used a different distillation loss as defined in Hinton et al. (2015) defined as:

$$L_D(t_o, s_o) = -\sum_{i=1}^{l} t_o^i log(s_o^i) \tag{3}$$

where l is the number of labels, t_o^i and $s_o i$ are the modified versions of the teacher model outputs and the current student model outputs. So the modified loss function during the incremental step for center loss is defined as:

$$L_{\text{incremental}} = l_{\text{metric learning}} + \lambda_{\text{distil}} * L_D, \tag{4}$$

where L_D is defined above, and the rest remains the same.

2.6 Loss Functions

The research focuses on catastrophic forgetting in metric learning methods, and as such, we consider three similarity learning loss functions:

- **Triplet loss** by Wang et al. (2014); Schroff et al. (2015) has been shown to learn good representations for determining image and video similarity Huo & van Zyl (2020). The triplet comprises an anchor ground truth image, a positive image and a negative image. The positive image belongs to the same identity as ground truth, and a negative image is selected from an identity that differs from the anchor Musgrave et al. (2020a).
- **Contrastive loss** learns optimal representations by using pairs of positive (matching class) and negative (non-matching classes) images. It functions by reducing the distance between positive pairs and increasing the distance between negative pairs in the embedding space Musgrave et al. (2020a).
- **Center loss** by Wen et al. (2016) learns a center for representations of unique classes. It simultaneously penalizes the distances between the representations of the images and their corresponding class centers that maximize inter-class separation and intra-class compactness. Center loss cannot be used directly as a loss function and is therefore paired with softmax.

3 Methodology

3.1 Datasets

All methods are subjected to incremental learning scenarios on well-known datasets to analyze the impact of catastrophic forgetting for metric learning. The datasets utilised are CUB200 Welinder et al. (2010) and Cars196 Krause et al. (2013). Ordinarily, the first half of the classes are used to train the network, while the second half is utilized for testing. For the CUB200 and Cars196 datasets, we used the first 120 classes as our base training class and 40 classes as incremental classes.

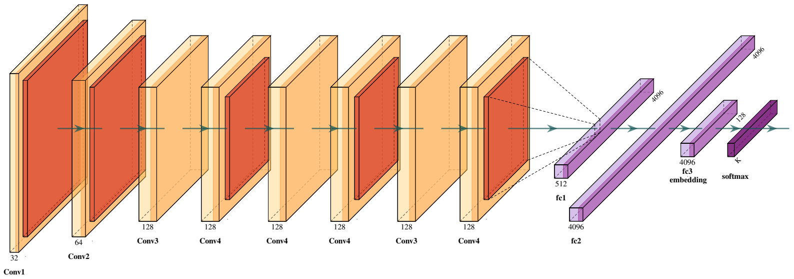

Figure 1: Convolution neural network Architecture. Yellow represents the convolution layers, orange the pooling layers, and purple the fully connected layers. K is the number of classes. The softmax fully connected layer is only used for Center Loss. For the other loss functions, we use the network up to fc3.

3.2 Architecture of CNN

We utilized a VGG-11 backbone deep neural network architecture. After the convolution layers were changed from the original 4096 to 512, the output layer was modified to 128 for similarity learning purposes. The network serves as an example of a commonly employed deep neural network architecture to assess the real-world performances of the respective methods more accurately. Our FullVAEs architectures consisted of 512 input layers (same size as out after the convolution layers), followed by 256, 128, and 128 for the bottleneck. Then these layers are reconstructed symmetrically in reverse. Our ConVAEs architecture consisted of a 2D Convolution layer with 512 input channels, followed by 2D Convolution with 32 input channels, followed by 2D Convolution with 16 input channels, and finally a bottleneck linear layer of input size 1600 and 256 output. These layers are reconstructed symmetrically in reverse so they can perform reconstructions with deConvolutional layers. All kernels for the de/Convolution layers are (3,3).

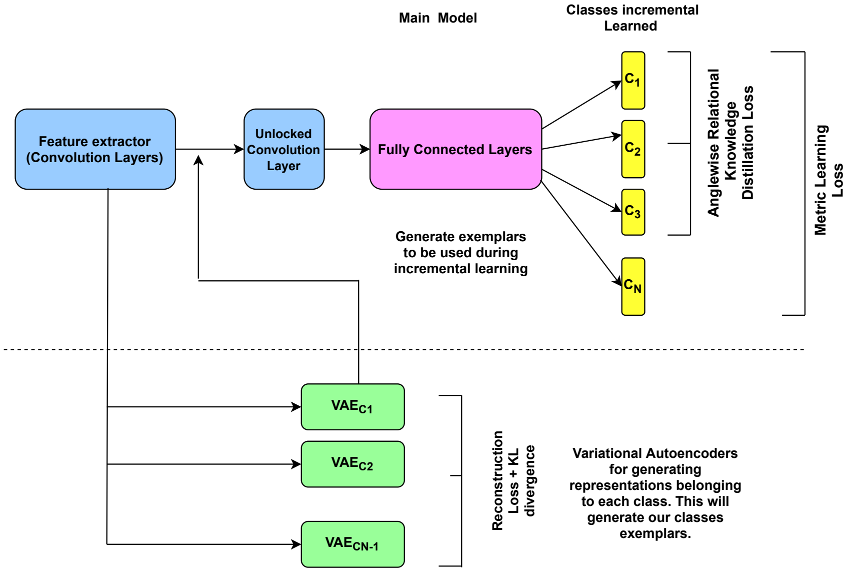
Main Model
Classes incremental Learned
Feature extractor (Convolution Layers)
Unlocked Convolution Layer
Fully Connected Layers
C_1
C_2
C_3
C_N
Anglewise Relational Knowledge Distillation Loss
Metric Learning Loss
Generate exemplars to be used during incremental learning
VAE_{C1}
VAE_{C2}
VAE_{CN-1}
Reconstruction Loss + KL divergence
Variational Autoencoders for generating representations belonging to each class. This will generate our classes exemplars.

Figure 2: The feature extractor is the convolutions layers of our VGG11 network. After initial training on the base-training set the feature extractor is frozen except the last set of convolutional layers Huo & van Zyl (2021).

3.3 Experimental Setup

Pre-processing done on the CUB200 and CARS196 datasets included normalizing the pixel values with mean and standard deviation of [0.485,0.456,0.406], [0.229,0.224,0.225] respectively for both datasets. Additionally, padding of size four and data augmentation by horizontal flipping when training was used. The additional padding and augmentation are used to overcome the dataset's complexity and, in so doing, obtain sufficient performance to demonstrate the negative effects of catastrophic forgetting.

Training and validation Pairs and Triplets were generated online. Triplets were mined employing semi-hard negative mining on positive and negative image pairs. To create pairs of images for contrastive loss, we performed pair margin mining. All mining was conducted using the Pytorch Metric Learning library Musgrave et al. (2020b) with the hyper-parameters stipulated below.

The positive and negative margins for contrastive lass were $[0, 0.3841]$ for CUB200 and $[0.2652, 0.5409]$ for CARS196. These margins were obtained from Musgrave *et al.*. For triplet loss, the margin for CUB200 and CARS196 were $[0.0961, 0.1190]$ respectively. The hyper-parameters [λ, α] for centre loss were [1.0, .5] respectively previously shown to have excellent outcomes for class separation Wang et al. (2017). We weighted the metric learning loss and distillation loss proportionately to the metric loss for iCaRL and our approach. The Adam optimizer was used with a learning rate of .001 was used for base set training and a learning rate of .0001 for incrementally learning new classes, β_1 value of .9, and β_2 value of .999. The Adam optimizer was used for iCaRL and FullVAE approaches. We needed to use the SGD optimizer for the ConvVAEr approach as we found it better for the method with the same learning rate as the Adam optimizer. For our method, we trained one variational autoencoder for each class that the network has seen for each incremental train step. The same Adam optimizer was used for training but with a constant learning rate of .001.

3.3.1 Exemplars for iCaRL

Rebuffi et al. (2017), used 2000 exemplars for CIFAR100, which makes available roughly 3% of the images to iCaRL. Due to the limited number of images per class for CUB200 and CARS196 (60 or fewer images per class) and substantially above previous work. We allow iCaRL to retain 480 exemplars of the total training images of both datasets which are roughly 5% for CARS196 and 6% for CUB200 of the total training images from 160 total classes. This percentage of retained exemplars is initially equivalent to 9% of the images per class and slowly decreases as additional classes are incrementally learned until the 5/6 % is reached. Therefore in our experiments, we limited the maximum number of exemplars for CUB200 and CARS196 to 480 exemplars for the 160 classes. Although we trained 120 of the 160 initially then incrementally learned 40 classes.

3.3.2 Training and Testing

We used a similar procedure for incremental learning to that of Kemker et al. (2018). We start with 120 of the classes from each of the datasets: CUB200 and CARS196. Subsequent training contains data from a single new unseen class up to 160 classes. A high-level overview of the steps, followed, are:

1. We take all the classes and split them into two sets of classes. A baseline set and the incremental set of classes.
2. We split the baseline set into base-training and base-test data sets using an 80/20 split.
3. We split the incremental set into inc-training and inc-test data sets using an 70/30 split.
4. We use the base-training and train our initial base models for incremental learning.
5. We take one unseen class from our incremental set of classes and previously last seen class to supplement the unseen class.
6. We retrain our base model with the inc-training data set for that unseen class.
7. We use the base-test data set from our baseline set to record the mAP@R after each step. We use base-test plus inc-test to record mAP@R for Ω_{all}.
8. We repeat from step 5 until all of the incremental sets' classes are exhausted.

Since metric learning loss functions require at least two classes, we take a single class from the previously learned data to pair with an all-new class. After training, we measure the mean average precision (mAP@R) on the new class to assess if the models are still learning. All the models were trained for 200 epochs for the base set training and 100 epochs for inc-set training. Due to the limited training samples in each class, we could not do early stopping with validation. However, we first experimented with validation during ”dummy” training to check for the best average epochs needed to reach decent performance before we started our experiments. We settled on 200 and 100 epochs for the base set and inc-set training, respectively.

We randomly split the data into two groups of classes consisting of a baseline set and an incremental set using three random seeds. Each run consisted of the same training and testing splits into the data to keep results consistent between all methods and we can average the performance.

The models' output is a feature representation of size 128 per image evaluated using mean Average Precision at R (mAP@R). Average precision at R (AP@R) is calculated using a single query identity used to retrieve the top R related relevant images from the database. The mAP at R is defined in the paper by Musgrave et al. (2020a).

4 Results and Discussion

The test metrics used are a minor variation to the incremental learning evaluation metrics of Kemker et al. (2018) to support learning based on distance rather than classification. We replace the classification accuracy metric with mAP@R to measure the performance as our models are not classifying images but instead retrieving them. We sequentially measured mAP@R on the base-test set after learning classes to keep track of base-test performance overall incremental steps. We tracked the model's performance on the new class by measuring the mAP@R performance of the new class on previously learned classes in the test data. We measured how well a model retains prior knowledge and learns new knowledge by measuring the mean mAP@R performance on the test data of all classes already learned during each training session. The metrics in the paper by Kemker et al. (2018) used are:

$$\Omega_{base} = \frac{1}{T-1}\sum_{i=2}^{T}\frac{\alpha_{base,i}}{\alpha_{ideal}}; \quad \Omega_{new} = \frac{1}{T-1}\sum_{i=2}^{T}\alpha_{new,i}; \quad \Omega_{all} = \frac{1}{T-1}\sum_{i=2}^{T}\frac{\alpha_{all,i}}{\alpha_{ideal}} \tag{5}$$

where T is the total number of training sessions, $\alpha_{new,i}$ is the test mAP@R for inc-test data of class i immediately after it is learned. $\alpha_{base,i}$ is the test mAP@R on the first session (base-test set) after ith new sessions have been learned. $\alpha_{all,i}$ is the test mAP@R of all of the inc-test data and base-test set for the classes seen so far. α_{ideal} is the offline model mAP on the base-test set, which is the ideal performance. Ω_{base} measures a model's retention of the base knowledge, after sequential training sessions. Ω_{new} measures the model's performance on new classes. Ω_{all} indicates how well a model both retains prior knowledge and acquires new information (how well we retrieve new learnt class among previously seen classes). Ω_{base} and Ω_{all} are normalized with α_{ideal}. The evaluation metrics are between [0,1] unless the results exceed the offline model. In our case we assume the α_{ideal} for the models are the averaged highest mAP@R performance that the respective models does initially on the base set to ensures it is fair when normalizing each methods performance by their respective best performances.

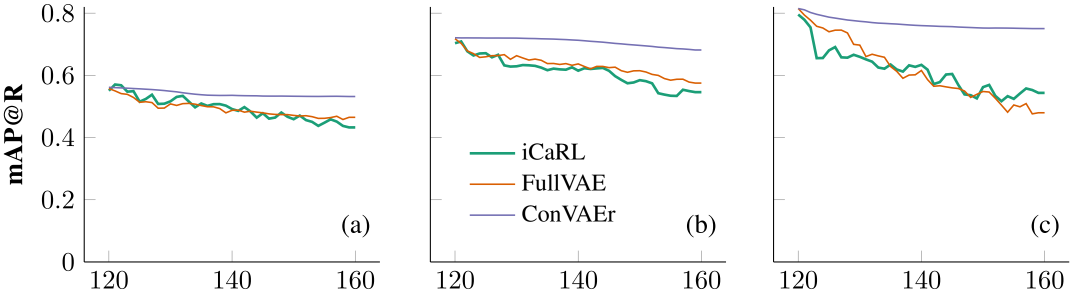

Figure 3: **CARS196**: mean average precision (mAP@R) on base-test for *(a) Triplet*, *(b) Contrastive*, and *(c) Center* loss combined for each incremental learning step.

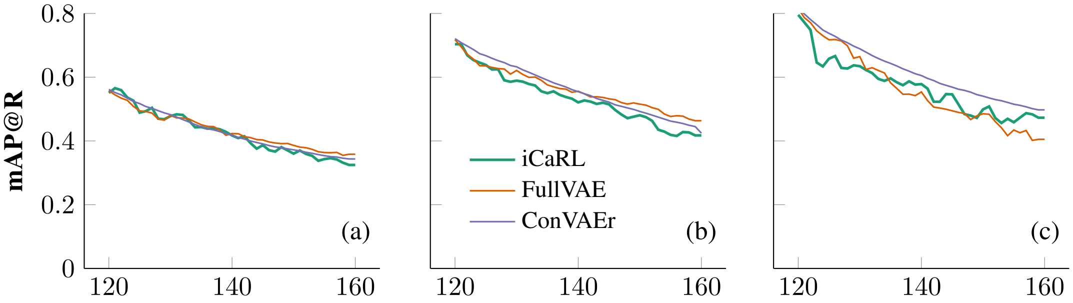

Figure 4: **CARS196**: mean average precision (mAP@R) on base-test plus inc-test for *(a) Triplet*, *(b) Contrastive*, and *(c) Center* loss combined for each incremental learning step.

4.1 Evaluation of results

Figures 3, and 4 highlight how each of the methods implemented reduces catastrophic forgetting during sequential class learning by testing on a base-test set after each new class is introduced. We observe that the FullVAE and ConVAEr retain better base knowledge compared to iCaRL during incremental learning. We further observe from Figures 3 that as progress with incremental steps, iCaRL performance drops quicker compared to the FullVAE and ConVAERr due to the number of samples for each class being reduced.

Table 1 shows the evaluation metric results utilising Equation 5 for all of the models. The values: Ω_{base}, Ω_{new}, and Ω_{all} span within $[0, 1]$. 0 suggests the model retains no knowledge, and 1 means it preserves all knowledge. The Ω_{new} results show the mAP@R performance on the test data of the newly learned class. The Ω_{all} presents how well the models recall prior knowledge and obtains new knowledge. The Ω_{new} results show the models are learning new knowledge very slowly and would not be useful as presented. In Table 1 we evaluated how methods retained previously and newly learned knowledge by testing on base-test set (old learned classes) and inc-test set (newly learned

Table 1: Incremental class test's mean average precision (mAP@R) starting from the memorised Ω_{base} with the new classes Ω_{new} added and the overall result Ω_{all}

			Ω_{base}			Ω_{new}			Ω_{all}	
Dataset	**Loss**	**iCARL**	**F'VAE**	**ConVAEr**	**iCARL**	**F'VAE**	**ConVAEr**	**iCARL**	**F'VAE**	**ConVAEr**
CUB-200	Contrast'	0.8461	0.9272	**0.9920**	0.2102	0.1299	0.1243	0.7380	0.7856	**0.7879**
	Triplet	0.8593	0.8651	0.9710	0.2282	0.2084	0.1300	0.7476	0.7531	0.7784
	Center	0.8008	0.7564	0.8380	0.3020	**0.3052**	0.0985	0.7235	0.6990	0.6761
CARS-196	Contrast'	0.8683	0.8800	**0.9814**	0.4148	0.3843	0.1911	0.7573	**0.7805**	0.7801
	Triplet	0.8952	0.8809	0.9623	0.3467	0.2961	0.1658	0.7683	0.7765	0.7615
	Center	0.7650	0.7499	0.9393	**0.5006**	0.4219	0.1620	0.7090	0.6879	0.6173

classes). The results are standardized with the offline models' ideal performance using Equation 5. The ideal' performances were obtained in the same way described earlier.

The results in Table 1 show our approach as the most robust over a long period of incremental class learning for retaining base knowledge on datasets. Even though ConVAEr is noisy and does not use actual images as exemplars, it can still preserve a class well during incremental learning. However, we observe that center loss does not perform as well as contrastive and triplet loss with VAEs exemplars and have difficulty learning new classes.

We observe ConVAEr is better than iCaRL in terms of overall knowledge retention, but iCaRL is better at learning new classes. We observe that center loss has the highest drop in base knowledge on FullVAE and iCaRL but they gain the highest amount of new knowledge. This supports our argument that we can preserve previous learned knowledge's embedding space by representing images in a representation that can be passed through intermediate layers and get similar or better performance compared to iCARL. The representation protects old occupied regions and forming new regions.

Finally, in Table 1 we see that contrastive loss retains the most base knowledge followed by triplet, and center loss as shown by Ω_{base} value. It was also observed that center loss does not work effectively with the FullVAE and ConvVAEr method as supported by the work of Huo & van Zyl (2021).

5 Discussion and future work

From the Ω_{new} results in Table 1, we can observe that iCaRL performed slightly better FullVAE when learning new classes but vastly outperform ConVAEr in some cases. In theory, ConVAEr should perform well for new classes as well with an additional unlocked convolutional layer in line with FullVAE. We theorize the VAE representation for simpler intermediate layers (fully connected layers) of the network were more accurate representations compared to the VAE representation for more complex intermediate layers (CNN layers), which are more difficult to replicate. The noisier VAE representations could be the cause of difficulty when learning new classes which we can observe from Table 1. Our speculation for the need for better VAE for ConVAEr derived from the center loss ConVAEr results that performed not so well on the new classes. For center loss, if the VAE representations are not closely generated around previously optimal centers and VAE representations cover too much space around the centers and cause difficulty learning new classes. We are required to perform more future work on this matter to investigate the issue.

6 Conclusions

We investigated the robustness ConVAEr method with contrastive, center, and triplet loss functions for catastrophic forgetting during incremental similarity learning and we have compared it to iCaRL and FullVAE. The results showed the effectiveness of each method on the three different loss functions against catastrophic forgetting. We found that contrastive loss with similarity pair mining performed the best against catastrophic forgetting on the base knowledge and performed the best average overall for base-test and inc-test data during incremental training followed by triplet loss. However, iCaRL is more effective when used with center loss. iCaRL is also more effective at learning new classes.

7 Reproducibility

For our efforts towards reproducibility of the results, we used python 3.6 across all machines and the same PyTorch libraries across both machines. We used seeds from one to three to random shuffle the 160 classes from the CARS196 and CUB200 datasets before taking the first 120 classes for base training and the remaining 40 classes in sequential order. This ensures that all methods start with the same classes and learn classes in the same order. We also ensured a random seed of 42 when splitting the training and test sets for training during incremental training to ensure that we are training and testing on the same splits across all the methods.

7.1 Hardware and software

We used two machines. An Intel(R) Xeon(R) CPU E5-2683 processor and an Intel(R) Core(TM) i7-5820K, both have 32 GB of RAM, and a GTX 1080 TI 11GB GPU. Both machines used Linux, Python 3.6, Pytorch 1.7.1 Paszke et al. (2019), Scikit-learn 0.23.2, and Pytorch Metric Learning 0.9.97.

8 Acknowledgements

We would like to acknowledge the Nedbank Research and Innovation Chair for supporting this research. The financial assistance of the National Research Foundation (NRF) towards this research is hereby acknowledged. Opinions expressed and conclusions arrived at, are those of the author and are not necessarily to be attributed to the NRF.

References

Rahaf Aljundi, Francesca Babiloni, Mohamed Elhoseiny, Marcus Rohrbach, and Tinne Tuytelaars. Memory aware synapses: Learning what (not) to forget. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 139–154, 2018.

Euntae Choi, Kyungmi Lee, and Kiyoung Choi. Autoencoder-based incremental class learning without retraining on old data. *arXiv preprint arXiv:1907.07872*, 2019.

Timothy J Draelos, Nadine E Miner, Christopher C Lamb, Jonathan A Cox, Craig M Vineyard, Kristofor D Carlson, William M Severa, Conrad D James, and James B Aimone. Neurogenesis deep learning: Extending deep networks to accommodate new classes. In *2017 International Joint Conference on Neural Networks (IJCNN)*, pp. 526–533. IEEE, 2017.

Chrisantha Fernando, Dylan Banarse, Charles Blundell, Yori Zwols, David Ha, Andrei A Rusu, Alexander Pritzel, and Daan Wierstra. Pathnet: Evolution channels gradient descent in super neural networks. *arXiv preprint arXiv:1701.08734*, 2017.

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.

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

Jiahao Huo and Terence L van Zyl. Unique faces recognition in videos. In *2020 23rd International Conference on Information Fusion (FUSION 2020)*, 2020. ISBN 978-0-964527-6-2.

Jiahao Huo and Terence L. van Zyl. Incremental class learning using variational autoencoders with similarity learning, 2021.

Ronald Kemker, Marc McClure, Angelina Abitino, Tyler L Hayes, and Christopher Kanan. Measuring catastrophic forgetting in neural networks. In *Thirty-second AAAI conference on artificial intelligence*, 2018.

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.

Jonathan Krause, Michael Stark, Jia Deng, and Li Fei-Fei. 3d object representations for fine-grained categorization. In *4th International IEEE Workshop on 3D Representation and Recognition (3dRR-13)*, Sydney, Australia, 2013.

Michael McCloskey and Neal J Cohen. Catastrophic interference in connectionist networks: The sequential learning problem. In *Psychology of learning and motivation*, volume 24, pp. 109–165. Elsevier, 1989.

Thomas Mensink, Jakob Verbeek, Florent Perronnin, and Gabriela Csurka. Distance-based image classification: Generalizing to new classes at near-zero cost. *IEEE transactions on pattern analysis and machine intelligence*, 35(11):2624–2637, 2013.

Kevin Musgrave, Serge Belongie, and Ser-Nam Lim. A metric learning reality check. In *European Conference on Computer Vision*, pp. 681–699. Springer, 2020a.

Kevin Musgrave, Serge Belongie, and Ser-Nam Lim. Pytorch metric learning, 2020b.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett (eds.), *Advances in Neural Information Processing Systems 32*, pp. 8024–8035. Curran Associates, Inc., 2019.

Robi Polikar, Lalita Upda, Satish S Upda, and Vasant Honavar. Learn++: An incremental learning algorithm for supervised neural networks. *IEEE transactions on systems, man, and cybernetics, part C (applications and reviews)*, 31(4):497–508, 2001.

Amal Rannen, Rahaf Aljundi, Matthew B Blaschko, and Tinne Tuytelaars. Encoder based lifelong learning. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 1320–1328, 2017.

Roger Ratcliff. Connectionist models of recognition memory: constraints imposed by learning and forgetting functions. *Psychological review*, 97(2):285, 1990.

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*, pp. 2001–2010, 2017.

Anthony Robins. Catastrophic forgetting, rehearsal and pseudorehearsal. *Connection Science*, 7(2): 123–146, 1995.

Florian Schroff, Dmitry Kalenichenko, and James Philbin. Facenet: A unified embedding for face recognition and clustering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 815–823, 2015.

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

Jian Wang, Feng Zhou, Shilei Wen, Xiao Liu, and Yuanqing Lin. Deep metric learning with angular loss. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 2593–2601, 2017.

Jiang Wang, Yang Song, Thomas Leung, Chuck Rosenberg, Jingbin Wang, James Philbin, Bo Chen, and Ying Wu. Learning fine-grained image similarity with deep ranking. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1386–1393, 2014.

P. Welinder, S. Branson, T. Mita, C. Wah, F. Schroff, S. Belongie, and P. Perona. Caltech-UCSD Birds 200. Technical Report CNS-TR-2010-001, California Institute of Technology, 2010.

Yandong Wen, Kaipeng Zhang, Zhifeng Li, and Yu Qiao. A discriminative feature learning approach for deep face recognition. In *European conference on computer vision*, pp. 499–515. Springer, 2016.

Friedemann Zenke, Ben Poole, and Surya Ganguli. Continual learning through synaptic intelligence. *Proceedings of machine learning research*, 70:3987, 2017.